50376
1987
297

N° d'ordre : 182

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

DOCTEUR EN ELECTRONIQUE

par

François BARANOWSKI

ETUDE DE LA CONCEPTION DE CONTROLEURS DE SECURITE EN LOGIQUE DYNAMISEE

EVALUATION PROBABILISTE DE L'INSECURITE RESIDUELLE

Soutenue le 4 Décembre 1987 devant la Commission d'Examen

Membres du Jury :	MM.	R.	GABILLARD	Président
		F.	LOUAGE	Rapporteur
		Y.	DAVID	Rapporteur
		V.	CORDONNIER	Examinateur
		J.F.	DHALLUIN	Examinateur
		B.	LE TRUNG	Examinateur

AVANT-PROPOS

Ce travail a été réalisé dans le Laboratoire de Radiopropagation et Electronique de l'Université des Sciences et Techniques de LILLE dirigé par M. le Professeur R.GABILLARD.

Qu'il trouve ici mes plus vifs remerciements pour m'avoir accueilli et sensibilisé aux problèmes de sécurité dont il est un des plus éminents spécialistes ainsi que pour l'honneur qu'il me fait en présidant ce jury.

Je remercie M. le Professeur V.CORDONNIER et M. le Professeur F.LOUAGE pour le grand honneur qu'ils me font en participant au jury de cette thèse.

J'exprime ma profonde reconnaissance à M. Y.DAVID, directeur de l'INRETS-CRESTA, pour les remarques qu'il a bien voulu formuler sur ce travail et pour l'intérèt qu'il y témoigne en le jugeant.

Je profite de l'opportunité qui m'est offerte pour remercier M. J-F.DHALLUIN, ingénieur à Matra Transport, pour ses conseils avisés et pour le temps qu'il a pu me consacrer durant mes recherches, j'apprécie l'honneur qu'il me fait en participant à ce jury.

Je remercie également M. B.LE TRUNG de l'INRETS-CRESTA pour l'intérèt qu'il me témoigne en participant lui aussi à ce jury, sa compétence en matière de sécurité étant connue de tous.

Enfin, je n'oublie pas de remercier l'ensemble du personnel du Laboratoire, M. C.HYPACE du CRESTA et tous ceux qui directement ou indirectement ont fait que ce travail s'est concrétisé par ce rapport.

INTRODUCTION GENERALE

CHAPITRE 1

POSITION DU PROBLEME

CHAPITRE 2

CONCEPTION DETERMINISTE

CHAPITRE 3

TEST DES CIRCUITS LOGIQUES

CHAPITRE 4

EXEMPLES DE CONCEPTION

INTRODUCTION GENERALE

Il est des domaines d'activité, particulièrement celui des transports collectifs automatisés auquel nous avons été sensibilisés, où la sécurité des personnes est prioritaire.

L'évolution technologique aidant mais aussi la demande accrue en performances de plus en plus grandes pour améliorer le confort des utilisateurs et des exploitants, les processus sont désormais gérés de manière numérique et le microprocesseur est entré pour une large part dans cette façon de concevoir une réalisation.

Afin de bénéficier de ses intéressantes propriétés, les recherches se sont orientées vers des méthodes permettant de détecter ses mauvais fonctionnements.

Notre travail s'inscrit dans le cadre de la sécurité liée à l'emploi du microprocesseur et où apparait la nécessité d'avoir des circuits extérieurs jugeant et décidant de la validité du comportement de ce composant.

Nous définissons des circuits de décision qui se doivent de présenter une grande sûreté de fonctionnement et donc de détecter leurs propres défaillances (autotest). Ces circuits ne sont pas uniquement liés à la microinformatique sécuritaire, mais nous avons limité notre étude à ce cadre qui est un domaine de recherche actuellement très actif.

Après avoir présenté, dans l'exposé de quelques principes utilisés pour détecter les défaillances des microprocesseurs, un aperçu des tâches fonctionnelles attendues et l'aspect sécurité des circuits contrôleurs, nous abordons dans un deuxième chapitre les points importants à maitriser dans une conception déterministe. Celle ci, par définition, prend en compte toutes les éventualités susceptibles de se produire durant le fonctionnement. L'appel à une technologie différente de ce qui est habituellement fait en la matière, les circuits logiques à faible niveau d'intégration, oriente cette revue des problèmes rencontrés.

Nous poursuivons ensuite sur le test de ces circuits logiques. Celui ci ne peut se concevoir que comme un test en ligne, seul capable d'assurer une sûreté optimale mais qui risque de présenter des carences. Nous définissons alors une méthode de calcul permettant d'estimer la part d'insécurité en résultant.

Le dernier chapitre est consacré à l'exposé de deux méthodes de mise en oeuvre de circuits logiques. La première, très structurée, présente un grand potentiel de détection, la seconde que nous développons au travers d'un exemple de réalisation est plus spécialisée dans l'analyse de signaux. Cette réalisation concrète, nous permet d'appliquer notre méthode de calcul et de chiffrer de manière probabiliste la sécurité ou l'insécurité du contrôleur obtenu.

INTRODUCTION

L'emploi de microprocesseurs dans la gestion de processus nécessitant un niveau de sécurité important est conditionné à l'utilisation de méthodes de test capables de détecter les défaillances de ces composants électroniques.

La variété des domaines d'activité, ferroviaire, avionique, nucléaire, etc..., et les cas particuliers présentés par les contraintes d'exploitation, de sécurité et de disponibilité imposent des architectures de systèmes différentes, des méthodes de test différentes (REF 1).

Vu l'attribution des fonctions, la défaillance de l'unité de gestion à microprocesseur (microprocesseur, mémoires, circuits associés) est une information qu'il faut traiter en priorité, en permanence et de manière sécuritaire.

Cette tâche est confiée à des structures de contrôle externes réalisées dans des technologies parfaitement maitrisées. Celles ci sont capables de reconfigurer l'architecture en vue d'une exploitation en mode dégradé ou d'arréter le processus avant qu'il ne devienne dangereux pour son environnement immédiat.

Ces circuiteries annexes que nous appellerons "contrôleurs" se différencient selon les méthodes de détection utilisées. Afin de dégager certaines constantes, il est intéressant d'examiner les grands principes employés pour s'assurer du bon fonctionnement de ces structures à microprocesseurs.

Parmi les méthodes utilisées pour "piéger" des défaillances de fonctionnement d'un ensemble, on distingue trois grandes familles :

- la redondance avec comparaison.
- le codage des informations manipulées par le dispositif.
- l'autotest dont le test comportemental.

L'examen succint de chacune conduit à la définition de contrôleurs fonctionnellement différents.

I.1) LA REDONDANCE (REF.2).

* PRINCIPE.

Cette méthode associe deux ou plusieurs unités à microprocesseurs remplissant chacune la même fonction; disposant de données d'entrée identiques, elles délivrent des sorties identiques. La non similitude du comportement : divergence d'une unité par rapport à l'autre ou aux autres traduit la défaillance de celle ci.

La détection de pannes s'effectue donc par la comparaison des sorties ou d'autres informations issues des unités et pouvant apporter des renseignements significatifs de l'identité des fonctionnements.

La figure I.1 donne la structure d'une architecture de rang 2, deux unités travaillent en parallèle sur un même vecteur d'entrée et délivrent chacune un vecteur de sortie qui sert après comparaison à l'élaboration du vecteur de sortie effectif de l'ensemble.

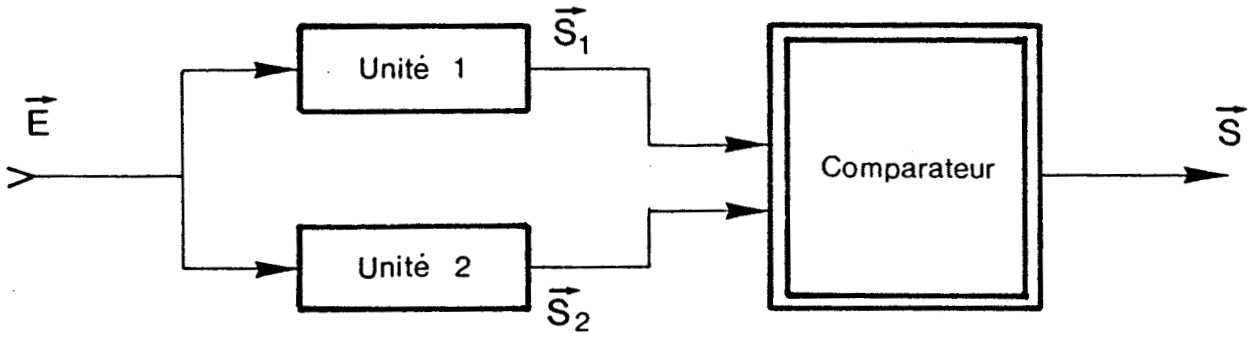


Fig. I.1

* CARACTERISTIQUES FONCTIONNELLES DU CONTROLEUR.

Le contrôleur a pour rôle de vérifier l'égalité des vecteurs $\vec{S}_1$ et $\vec{S}_2$ issus des unités redondantes.

$\forall \vec{S}_1$ et $\forall \vec{S}_2$ si $\vec{S}_1 = \vec{S}_2$ la sortie $\vec{S} = \vec{S}_1$ ou $\vec{S}_2$

si $\vec{S}_1 \neq \vec{S}_2$ la sortie $\vec{S} \neq \vec{S}_1$ et de $\vec{S}_2$

$\vec{S} = \vec{S}_s$

L'égalité des deux vecteurs $\vec{S}_1$ et $\vec{S}_2$ produit une sortie $\vec{S}$ identique à $\vec{S}_1$ et $\vec{S}_2$.

L'inégalité bloque les deux vecteurs, le comparateur délivre alors une sortie $\vec{S}_s$ dite " vecteur de sortie sécurité ", imposant ainsi au processus commandé, un état de sécurité tel que celui ci ne puisse nuire à son environnement.

Le choix de cet état dépend de la nature du processus à gérer :

- arrêt progressif
- arrêt partiel
- arrêt immédiat.

La complexité du contrôleur dépend des facteurs suivants :

- Rang de la redondance :

Il est possible en effet, de comparer plus de deux unités, par exemple trois, et d'effectuer un vote de type majoritaire tel que si l'une des unités diverge (dans ce cas on admet que les deux autres sont correctes), le processus puisse être piloté avec seulement deux unités en redondance de rang 2, la troisième étant déconnectée.

Le contrôleur est alors appelé voteur.

- Redondance hétérogène :

La sécurité de cette solution repose sur le fait qu'il n'existe pas de défaillance affectant de manière similaire plusieurs unités redondantes. Si une telle défaillance existe, elle ne peut être détectée (informations fausses et identiques).

La redondance hétérogène associe des unités technologiquement différentes pouvant fonctionner avec des horloges non synchrones, des logiques différentes afin de se prémunir de défaillances qui affecteraient de façon identique plusieurs unités.

Le contrôleur chargé de la comparaison doit pouvoir s'adapter en mémorisant et/ou en transformant certains vecteurs de sortie pour rendre la comparaison possible.

* ASPECT SECURITE DU CONTROLEUR.-

Si les défaillances de mode commun sont rendues impossibles (c'est d'ailleurs là toute la difficulté de la méthode), la sécurité est assurée par le contrôleur. Celui ci doit être conçu de manière à ce que toute panne qui l'affecte (et qui peut donc rendre sa fonction de comparaison plus permissive) impose le vecteur de sortie $\vec{S}_s$.

Le processus est alors mis en sécurité par excès.

I.2) LE CODAGE (REF.3-4).

* PRINCIPE.

La transmission d'un signal à travers un canal quelconque, nécessite l'emploi d'un codage à l'émission et d'un décodage à la réception afin de détecter (voire corriger) les perturbations induites par ce canal sur l'information.

Le traitement d'informations par microprocesseur peut être considéré en suivant une démarche similaire d'où, la notion de détection d'erreurs par codage.

Le processus est alors analogue à un canal de transmission associé à des transformations attendues de l'information (fonctions réalisées par ce microprocesseur) et à des transformations non attendues qui sont générées par les défaillances.

Le codage de l'information avant traitement permet à postériori de révéler l'altération de cette information par la non appartenance au code en sortie.

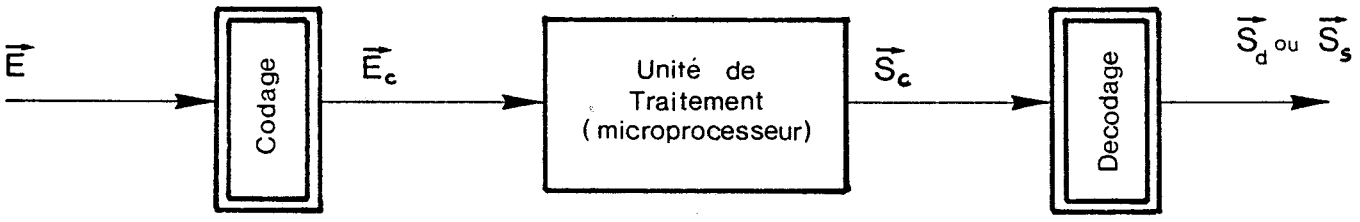


Fig. I.2

* ASPECTS FONCTIONNELS DU CONTROLEUR.

Dans ce cas de figure, il s'agit essentiellement de créer des unités de codage et de décodage appropriées au code choisi pour tester le microprocesseur, la fonction de contrôle revenant au décodeur.

Le choix du code dépend :

- des modes de défaillance à détecter
- du niveau de sécurité attribué au dispositif (nombre de bits de redondance)
- du type des manipulations effectuées par le microprocesseur sur l'information.

Le décodeur vérifie l'appartenance du vecteur de sortie codé au code attendu, il transmet une sortie décodée $\vec{S}_d$ ou déconnecte l'unité microprocesseur et impose un vecteur sécurité $\vec{S}_s$ prédéterminé.

* ASPECT SECURITE.

L'aspect sécurité est ici tout aussi important, codage et décodage doivent être effectués de façon sécuritaire en relation avec l'objectif de sécurité visé.

Toute défaillance du décodeur et du codeur, doit conduire à la génération du vecteur de sortie sécuritaire S_s.

I.3) TEST COMPORTEMENTAL (REF.6).

* PRINCIPE GENERAL.

Ces tests sont basés sur l'observation de la bonne exécution des instructions élémentaires logiques, arithmétiques et de manipulation de données du microprocesseur.

Le résultat de l'observation doit permettre de générer une information exploitable de l'extérieur (exemple typique du signal de chien de garde) et rendant compte du résultat des tests efffectués.

On peut examiner plus particulièrement une de ces méthodes, celle ci donnera lieu, au dernier chapitre, à l'analyse plus détaillée de son contrôleur (REF.5).

La méthode utilise pour tester le microprocesseur deux principes :

- la mesure de la durée d'exécution du logiciel d'application
- la réponse des instructions élémentaires à des stimulations déterministes (REF.7).

L'ensemble des résultats est traduit par un signal appelé signal équitemps, généré en permanence et dont les caractéristiques dynamiques sont les suivantes :

- fréquence fixe, f_o
- rapport cyclique, $r = 1/2$.

f_o et r traduisent le bon fonctionnement du microprocesseur si les valeurs constatées sur ces deux paramètres sont celles prévues (REF.8).

L'architecture avec le contrôleur est donnée figure I.3.

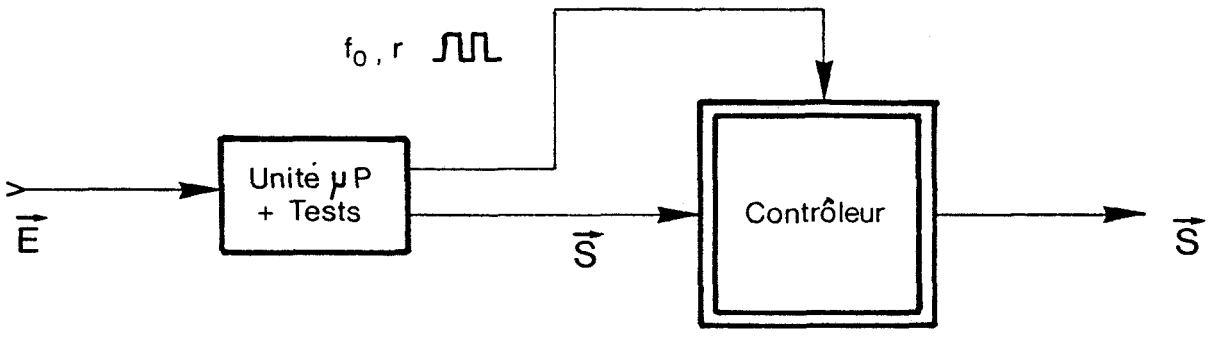


Fig. I.3

* ASPECTS FONCTIONNELS DU CONTROLEUR.

Le travail du contrôleur est lié à la nature de l'information, aux caractéristiques à traiter; dans notre cas, il s'agit de vérifier la fréquence et le rapport cyclique d'un signal. Les valeurs de ces deux paramètres conditionnent la validation du vecteur de sortie du microprocesseur sous test.

Il va de soi que plus l'information est simple, plus le contrôleur sera simple dans sa structure.

* ASPECT SECURITE.

Le contrôleur doit effectué son travail d'analyse de façon sûre et assurer que le signal de sortie sécuritaire S_S est effectivement imposé au processus lorsqu'une défaillance en son sein risque d'altérer sa capacité de contrôle.

I.4) CAHIER DES CHARGES D'UN CONTROLEUR.

Chacune des méthodes nécessite la présence d'une circuiterie annexe, au demeurant simple par rapport à la complexité du microprocesseur, pour effectuer l'opération de déconnection de celui ci s'il est défaillant et assurer ainsi la mise en sécurite du processus.

Il n'est pas envisageable, pour l'instant, de se passer de ces circuits de contrôle extérieurs au microprocesseur.

Nous allons résumer l'ensemble des points définis dans les exemples de contrôleurs examinés.

I.4.1) ASPECTS FONCTIONNELS.

Un contrôleur assure un certain nombre de tâches dont quelques unes sont directement liées à la nature de la méthode de détection adoptée.

- Extraction de l'information

Le contrôleur appréhende le vecteur de sortie de l'unité sous test et met en forme l'information susceptible de contenir le défaut.

- Traitement

Le traitement consiste à effectuer une discrimination simple, de type binaire : panne détectée ou bon fonctionnnement, éventuellement, ce traitement peut être plus complexe dans le cas de redondances d'ordre supérieur à 2 (localisation de l'unité défaillante).

- Actionnement des sorties

L'information binaire conditionne la transmission du vecteur de sortie issu du processeur ou au contraire impose un vecteur de sortie prédéterminé et sécuritaire S_s positionnant le processus dans un état non dangereux pour son environnement.

- Autres

Des fonctions annexes peuvent être créées telles que :

- la génération d'alarmes
- la mémorisation des défaillances en vue d'une maintenance plus rapide
- la commutation vers des unités en attente.
- etc...

I.4.2) ASPECT SECURITE.

Le report d'une partie de la sécurité de l'ensemble sur le contrôleur implique que celui ci possède une haute sûreté de fonctionnement, l'apparition de défaillances doit être maitrisée correctement. Nous pouvons examiner les conséquences de ces défaillances sur le processus lorsqu'elles apparaissent sur le processeur et/ou sur le contrôleur.

* Processeur Défaillant - Contrôleur non Défaillant

L'évènement est naturel et non dangereux, le contrôleur prend le processus en charge par l'intermédiaire du vecteur sécurité .

* Processeur non Défaillant - Contrôleur Défaillant

Si aucune précaution n'est prise sur ce contrôleur, il peut apparaitre :

- une altération, par ce contrôleur défaillant, du vecteur issu du processeur; le processus n'est plus maitrisé.
- le vecteur de sortie du processeur n'est pas altéré, mais lorsqu'une défaillance l'affectera, il se peut que le contrôleur ne soit pas alors en mesure de la détecter, la situation deviendra dangereuse (panne latente sur le contrôleur).

* Processeur Défaillant - Contrôleur Défaillant

Cet état du système fait suite à une panne latente du contrôleur ou du processeur. La situation est dangereuse dans la mesure où il n'existe plus de moyen de contrôler le processus.

Il apparait que la sécurité repose sur la conception du contrôleur (maitrise des conséquences des défaillances) mais aussi sur le fonctionnel puisqu'il faut connaitre tous les défauts que peut générer le processeur en panne.

I.5) CONCLUSION.

Le contrôleur constitue en lui même un maillon important pour la sécurité du processus.

Quelle que soit la solution retenue pour tester le processeur, il est nécessaire d'avoir en fin de chaine un élément de contrôle et de décision indépendant.

La conception des contrôleurs doit faire l'objet d'un soin tout particulier : aspect sécurité.

Il est difficilement envisageable d'appliquer à cette fonction les techniques de test comportemental ou de redondance. Empêcher l'apparition des pannes n'étant pas possible, la solution consiste alors à les prendre en considération dans la conception.

L'apparition d'une défaillance doit être autodétectée et conduire invariablement le point de fonctionnement du contrôleur vers un état unique correspondant à l'état de sécurité à imposer au processus.

Cette démarche parfaitement déterministe conduit à l'élaboration de contrôleurs sûrs, elle est synonyme de sécurité intrinsèque et, c'est vers ce genre d'approche que nous allons orienter la suite de ce rapport.

INTRODUCTION

Notre but est la conception de contrôleurs satisfaisant un cahier des charges fonctionnel donné tout en essayant d'atteindre une sécurité de fonctionnement qualifiable d'"absolue".

La sécurité dite absolue est déjà atteinte couramment sur les systèmes ferroviaires, par exemple, en faisant appel à des composants électriques et électroniques discrets connus de façon exhaustive et assemblés selon des règles qui sont devenues des règles de l'art.

Cette conception est-elle encore adaptée aux systèmes microinformatisés à controler ?

Nous nous proposons d'étudier les performances de contrôleurs conçus autour de circuits logiques intégrés simples qui semblent plus homogènes à la technologie des microprocesseurs.

Dans ce chapitre, nous examinons quelques points importants qui agissent sur la conception de contrôleurs de sécurité sur des bases déterministes.

D'une part le principe de sécurité intrinsèque est défini, ainsi que la manière avec laquelle il est appliqué actuellement ,même dans les réalisations les plus récentes, puis à ce niveau nous introduisons les circuits intégrés logiques pour en évaluer les avantages.

D'autre part, en replaçant le contrôleur dans son contexte, entre le microprocesseur et les actionneurs, nous examinons les contraintes à respecter pour s'assurer de la validité des informations reçues, traitées et générées.

CONCEPTION DETERMINISTE.

Au contraire d'une sécurité basée uniquement sur la fiabilité de constituants choisis et de la confiance que l'on peut leur accorder, la sécurité intrinsèque a une base parfaitement déterministe.

Un ensemble ou un sous ensemble répond aux principes déterministes, s'il est connu de façon exhaustive dans toutes ses réactions qu'elles soient normales ou induites par des défaillances et, que ces réactions ne conduisent pas à un ou des évènements jugés dangereux (REF.9).

II.1) SECURITE POSITIVE - SECURITE INTRINSEQUE.

Un système peut être qualifié de système de sécurité, si tout évènement anormal dont il est le siège provoque une réaction saine (état de sécurité). Le système est alors moins permissif qu'en bon fonctionnement.

Pour parvenir à ceci, on peut appliquer le principe de la sécurité positive où l'état de sécurité correspond à l'état de plus basse énergie du système (REF.10).

Par exemple, pour un véhicule terrestre cet état correspond à l'arrêt sur un terrain plat, moteur coupé.

Les informations transitant dans une chaine conçue en sécurité positive sont du type binaire :

- absence d'énergie = information 0
- présence d'énergie = information 1

L'application de ce concept passe par la conception de circuits consommant de l'énergie .

Le concept de sécurité intrinsèque, plus large quant aux moyens d'aboutir à un système de sécurité lui est préféré.

La conception d'un circuit en sécurité intrinsèque repose sur la notion de schéma. Un agencement particulier des composants élémentaires permet d'aboutir à ce qu'en cas de panne (quelle qu'elle soit) les sorties tendent toujours vers l'état de sécurité.

II.2) CONCEPTION EN SECURITE INTRINSEQUE.

Pour aboutir, il est nécessaire d'avoir une connaissance très poussée des composants utilisés et donc d'en déterminer les modes de pannes possibles mais aussi les modes de pannes impossibles sur lesquels pourra s'appuyer la conception.

Une étude de sécurité consiste alors à appliquer à la circuiterie toutes les défaillances possibles et à vérifier que les sorties tendent en toutes circonstances vers l'état de sécurité défini.

On imagine aisément qu'il n'est pas possible de réaliser de cette manière et brutalement des fonctions de grande importance. Par contre ceci est parfaitement envisageable pour des fonctions élémentaires binaires qui après assemblage selon des règles rigoureuses aboutiront à des fonctions logiques complexes et sécuritaires.

Actuellement, cette démarche est encore couramment utilisée et il faut reconnaitre qu'elle est d'une sûreté que l'on peut qualifier d'absolue. Cependant, ce type de réalisation souffre de par nature de quelques inconvénients :

- encombrement :

malgré l'utilisation de circuits du type à montage de surface, la réalisation de fonctions d'une certaine complexité prend encore beaucoup trop de place, il faut en effet recréer chaque porte logique comprenant en particulier des transformateurs inhérents à cette conception, ceci ne fait qu'accroitre la disproportion, on imagine alors difficilement un microprocesseur construit selon cette technique.

- suivi des composants :

il est nécessaire de s'assurer que les composants utilisés possèdent les mêmes caractéristiques de pannes et qu'un changement de constructeur ou de processus de fabrication ne les a pas modifiées, la refonte des schémas serait alors à prévoir.

- faibles performances dynamiques

- écart de génération très important avec les microprocesseurs

Il semble donc intéressant de trouver d'autres moyens pour aboutir à la conception de contrôleurs en sécurité intrinsèque sans les inconvénients mentionnés (REF.11).

Les critères de choix sont les suivants :

- technologie plus récente

- faible encombrement et faible consommation par rapport aux fonctions réalisées

- relative indépendance des modes de pannes par rapport aux constructeurs et aux filières de fabrication

- complexité suffisamment faible pour appréhender tous les modes de pannes facilement.

Le choix s'est porté naturellement sur les circuits intégrés logiques à très faible niveau d'intégration que sont les SSI et les MSI (small scale integration, medium scale integration).

II.3) CIRCUITS SSI ET MSI.

Ces circuits connus depuis les années soixante existent dans de très nombreuses technologies et font d'ailleurs encore l'objet d'améliorations des performances : fiabilité,consommation, rapidité....

L'intégration est très faible, de quelques portes à quelques dizaines de portes élémentaires par circuit intégré (REF.12). L'analyse du fonctionnement et des pannes qui peuvent survenir est encore, de ce fait, relativement aisée.

La modélisation que nous avons adoptée se situe au niveau de la porte logique élémentaire : portes OU, ET, NON, etc.... Dans l'analyse que nous faisons une fonction quelconque prise dans l'éventail disponible est décomposée en portes élémentaires reliées (Annexe 1).

Un des inconvénients de ces circuits vient d'une connectique importante (soudures) et on constate que la pluspart des pannes proviennent effectivement de là, elles seront prises en compte dans la sécurité: notre principal soucis. Le problème de la disponibilité sera résolu d'une part en concevant des contrôleurs les plus simples possibles et d'autre part grâce à une relative indépendance des défauts par rapport à la technologie par l' emploi éventuel de circuits intégrés personnalisés (custom ou semi-custom).

II.3.1) DEFAUTS SUR LES PORTES (REF.13).

Le niveau de modélisation a été choisi en faisant abstraction, dans la mesure du possible, de la technologie.

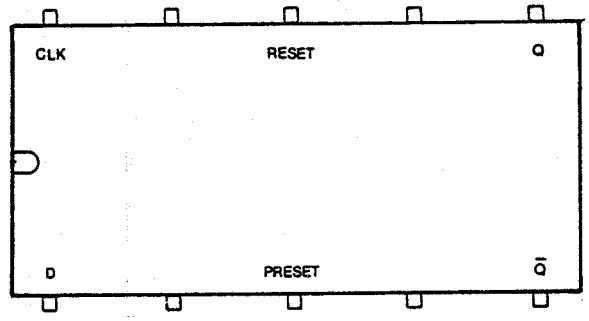
Niveau 0 : Le boitier avec ses entrées et ses sorties.

Niveau 1 : Il est constitué de plusieurs fonctions identiques qui caractérisent le boitier (combinatoire complexe, séquentiel...).

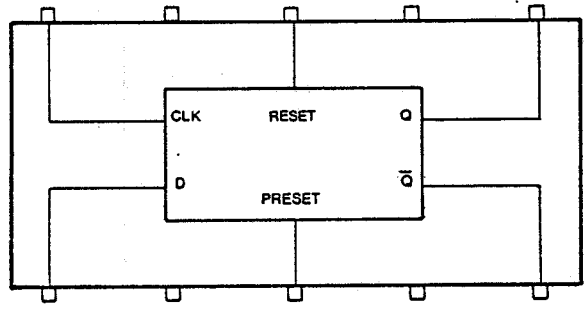
Niveau 2 : Portes logiques reliées réalisant la fonction.

Niveau 3 : La fonction est décrite à l'aide de composants élémentaires (transistors, résistances, etc...)

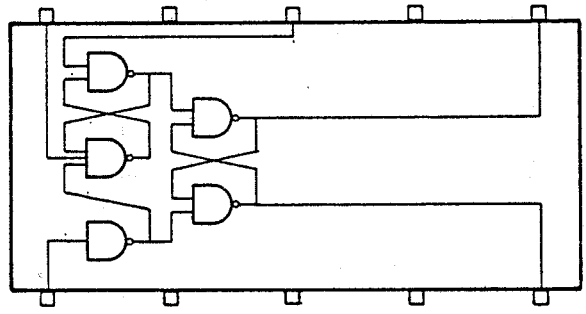
Niveau 0



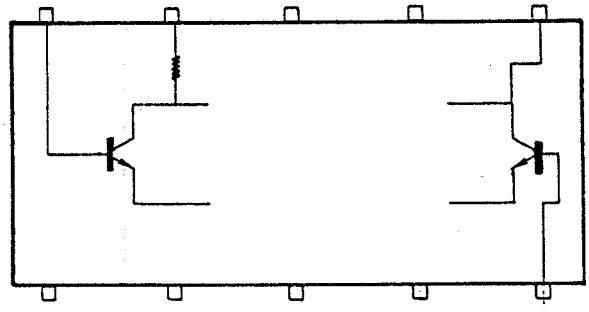
Niveau 1



Niveau 2



Niveau 3



Les défaillances prises en considération sont les suivantes :

Une panne physique peut être déterminée à partir du niveau 3, ses conséquences sont modélisées au niveau 2.

Exemple : Niveau 3 --> ouverture du collecteur d'un transistor
Niveau 2 --> collage de la sortie à la valeur 0

Un grand nombre de pannes au niveau 3 provoque des conséquences identiques vu du niveau 2 et donc, pour l'utilisateur, le détail du niveau 3 n'apportera pas de précisions intéressantes.

Une porte logique est considérée comme un élément isolé géographiquement et nous ne prendrons pas en compte les défaillances de type court-circuit entre composants internes de deux portes différentes. Les seules défaillances les mettant en liaison ont pour origine les connexions extérieures de ces portes : entrées, sorties .

Chaque technologie possède, bien entendu, des types de panne particuliers mais, en considérant une modélisation au niveau 2, on peut en faire abstraction et rendre l'étude très générale.

Les défauts constatés au niveau d'une porte élémentaire sont les suivants :

* les collages de sortie à 0 ou 1
* les collages ouverts : sortie équivalente à une haute impédance
* les retards à la montée et/ou à la descente du signal de sortie
* les effets mémoire
* les niveaux douteux en sortie : tension délivrée comprise entre les seuils admis du 0 et du 1 logique
* le changement de la fonction combinatoire normalement remplie par la porte considérée.

Ces anomalies, communes à toutes les technologies, admettent une modélisation logique, on peut ainsi recréer un circuit nouveau qui est alors analysé par les méthodes classiques de description des circuits binaires, on observe ainsi facilement le comportement du contrôleur affecté par la panne.

On prend également en considération les pannes mettant en jeu plusieurs portes en distinguant :

* les courts-circuits entre sorties de portes
* les erreurs injectées par l'entrée défaillante d'une porte aux autres portes qui lui sont connectées
* les ruptures de liaison entre portes

Ces trois dernières pannes sont également modélisables en logique.

Une démonstration de sécurité d'un contrôleur utilisant ces circuits consiste donc à appliquer systématiquement au niveau de chaque porte l'ensemble de ces défauts et à observer les conséquences sur le fonctionnement.

II.3.2) CONCLUSION (REF.11).

La faible complexité des circuits SSI et de certains MSI permet à priori une analyse exhaustive des modes de pannes qui peuvent les affecter.

La relative indépendance technologique vis à vis des fautes modélisées laisse présager qu'une fois la réalisation figée, il ne sera plus obligatoire de faire un suivi très strict chez le fabricant de composants : les fonctions logiques sont stables dans leur schéma, d'un fabricant à un autre , le remplacement pur et simple des composants par un équivalent est possible.

L'introduction de ces circuits dans la conception de contrôleurs de sécurité va , bien entendu, s'accompagner de problèmes spécifiques, le test en ligne en est un et il est examiné dans le chapitre suivant , d'autres, liés directement à l'environnement immédiat du contrôleur sont examinés aux paragraphes suivants.

II.4) ENVIRONNEMENT DU CONTROLEUR.

Le contrôleur communique avec le ou les microprocesseurs qu'il contrôle et avec les actionneurs qu'il alimente directement ou indirectement selon le résultat du contrôle.

Fig.II.1.

Le type d'information qu'il reçoit est important non seulement pour la sécurité du processus mais, également, pour la définition des moyens à mettre en oeuvre pour s'assurer que ce contrôleur fonctionne correctement, en accord avec les principes de sécurité définis en début de ce chapitre.

II.4.1) NATURE DES INFORMATIONS.

Selon la méthode de détection retenue pour les défaillances du processeur, la nature des informations reçues par le contrôleur est différente et on peut faire le classement suivant :

- les informations à caractère déterministe

- les informations à caractère aléatoire

Les unes et les autres sont directement impliquées dans la manière d'envisager le contrôleur.

II.4.1.1) INFORMATIONS DETERMINISTES

Le cas représentant parfaitement une information déterministe est celui du signal équitemps, celui ci est prévisible : fréquence et rapport cyclique connus.

Le concepteur connait alors, à chaque instant dans son contrôleur la forme des signaux élaborés, il peut déterminer également les transformations engendrées par des défaillances et constater qu'elles sont détectées ou non et, en ce cas il corrige sa réalisation.

On trouve également un bon exemple d'informations déterministes dans les processus ayant un fonctionnement cyclique, les informations se succèdent dans un ordre bien défini qu'il est possible de prévoir par construction.

A priori, la conception de contrôleurs manipulant de telles informations semble grandement facilitée par le caractère prévisible de celles ci ; elles sont directement en accord avec les principes déterministes.

II.4.1.2) INFORMATIONS A CARACTERE ALEATOIRE.

Cette caractéristique s'oppose aux informations précédentes. Si le processus évolue de manière cohérente, c'est ce qu'on attend de lui, les vecteurs de sortie n'ont pas un caractère cyclique.

Le contrôleur reçoit donc un flot d'informations sans lien direct entr'elles , il est alors difficile de prévoir ce que va être le cheminement de celles ci dans sa circuiterie et évaluer l'effet d'une défaillance sur la tâche de contrôle.

Il se peut que certaines ne soient révélées que pour des configurations particulières du processus donc, des vecteurs de sortie controlés.

Des informations de ce genre ne peuvent être manipulées directement, il est nécessaire de les transformer de façon à ce qu'elles acquièrent le déterminisme qui les rend plus faciles à manipuler. La logique alternative résoud ce problème (Cf chap.4).

Un exemple typique de contrôleur pouvant recevoir ce genre d'informations aléatoires est celui des comparateurs ou des voteurs , le contenu de l'information n'a de sens que pour une comparaison entre vecteurs issus des processeurs redondants.

II.4.2) SUPPORT DE L'INFORMATION.

L'aspect physique de l'information est particulièrement important pour la confiance que l'on peut accorder au contrôleur en cas de panne.

On peut rechercher le support le plus intéressant pour véhiculer une information à travers un circuit avec une sécurité suffisante. On distingue ainsi et entre autres :

* la forme statique
* la forme énergisée
* la forme dynamisée

II.4.2.1) FORME STATIQUE (REF.14).

C'est une façon simple de présenter l'information binaire :

- le niveau 0 de tension représente l'état repos de la variable

- le niveau 1 représente l'état travail

Les qualités de réaction en cas de pannes sont médiocres, même pour celles réputées fréquentes (collages, rupture de liaison électrique).

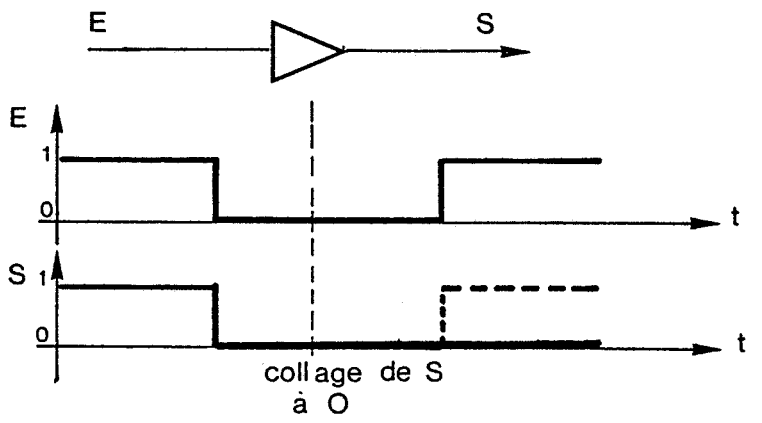


Fig.II.2

Dans le cas de la figure II.2 , l'état de la sortie S ne permet pas d'affirmer si le 0 est dû effectivement à la valeur à l'entrée de la porte ou à un collage de cette sortie à la valeur 0.

La sécurité de l'information n'est pas assurée, la forme statique ne peut être retenue comme support sûr.

II.4.2.2) FORME ENERGISEE.

Ce support répond au concept de sécurité positive et, pour avoir l'efficacité recherchée, les circuits doivent consommer l'énergie véhiculée par le signal.

Les circuits logiques utilisés ne consomment pas suffisamment, par leurs entrées normales, pour que ce support puisse être efficace.

Eventuellement, il serait possible d'exploiter ce concept en se servant des entrées d'alimentation en énergie des circuits logiques ou encore en utilisant des circuits du type collecteur ouvert mais ceci ne permettrait pas encore d'accéder aux fonctions complexes et sécuritaires.

Il faut donc à priori rejeter ce support d'information.

II.4.2.3) FORME DYNAMISEE (REF.14).

La dynamisation de l'information s'avère être une technique très commode pour détecter immédiatement des défaillances matérielles sur des circuits logiques.

Elle consiste à appliquer un codage simple ,ou modulation, sur les informations.

A chaque état d'une variable (0 et/ou 1), on associe une fréquence donnée. La figure II.3 en donne deux exemples.

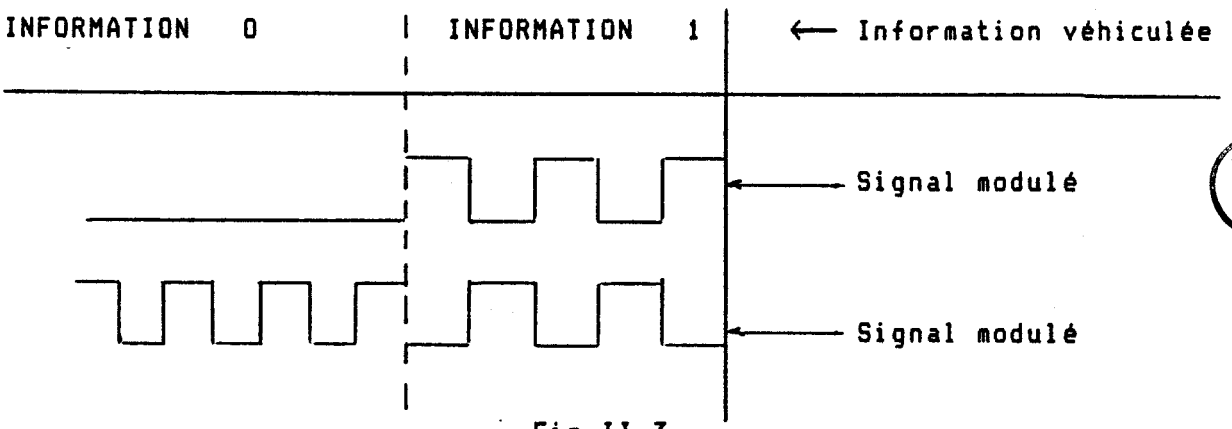


Fig II.3

On trouve également avec la logique alternative (figure II.4) une forme dynamisée très structurée : la valeur de la variable dynamisée est repérée par une horloge.

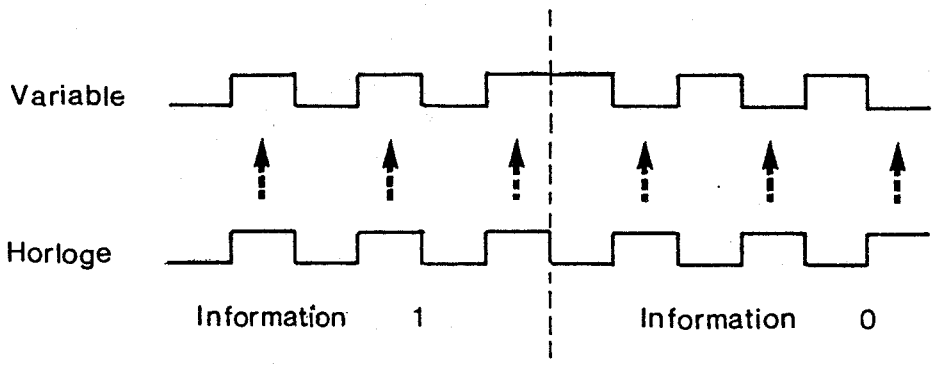


Fig II.4

En provoquant la dynamisation des informations, une plus grande partie des circuits traversés est testée. Si le circuit est conçu pour tenir compte de cette dynamisation au niveau de chaque fonction logique, il est facile de s'apercevoir d'une défaillance en un point quelconque par l' absence de dynamisation ou par son incohérence.

La forme dynamisée comme support d'information sera donc retenue dans la suite de ce rapport comme le moyen pour aboutir dans la conception.

II.4.2.4) CONTRAINTES ET LIMITES DE LA DYNAMISATION.

La dynamisation permet, en comparaison avec les deux autres supports et dans le cas de l'utilisation de circuits logiques, une détection à priori plus efficace et donc une sûreté de l'information et de ses manipulations plus grande.

Certaines contraintes sont toutefois évidentes :

* l'information émise par la source doit déjà être dynamisée. En effet, le microprocesseur sous contrôle doit assurer, lui même, par logiciel, la mise en forme des informations à destination du contrôleur. Ceci alourdit considérablement son travail dans la mesure où c'est un rafraichissement périodique qui constitue la dynamisation. C'est toutefois, le seul moyen d'avoir à l'entrée du contrôleur des informations ayant une certaine crédibilité.

* la dynamisation ne peut pas être anarchique, il est nécessaire de tenir compte de la période d'évolution du processus :

- plus le processus est rapide et plus la dynamisation doit être de fréquence élevée.

- la fréquence de dynamisation est un multiple de la fréquence d'évolution du processus (synchronisme).

Malgré ces remarques et en excluant le codage ,au sens classique du terme (parité, code k parmi n, etc...) qui n'entre pas directement dans notre propos, le support de l'information que constitue la forme dynamisée semble être le moyen le plus efficace et le plus simple pour transmettre en sécurité des ordres de commande compte tenu des défaillances des circuits manipulant ces informations.

II.4.3) PROBLEME POSE PAR LES SORTIES SECURITAIRES.

Le contrôleur, qui reçoit des informations, doit également en émettre vers son environnement. Le problème de la sûreté de celles ci est particulièrement important dans la mesure où par sa fonction ,il constitue le dernier élément de la chaine avant les actionneurs.

La transmission d'une information fausse est à juste titre un évènement catastrophique pour le processus.

Le choix, pour transmettre en sécurité des signaux compte tenu des remarques des paragraphes précédents, est restreint :

utilisation de la dynamisation.

Deux cas se présentent alors :

* le récepteur connecté à la suite du contrôleur peut être alimenté par des informations dynamisées et la solution est immédiate.

* le récepteur ne peut recevoir que des informations de type statique binaire "tout ou rien" et dans ce cas, il est nécessaire de procéder à la conversion par des circuits classiques de sécurité intrinsèque, les circuits logiques intégrés étant incapables de manipuler de façon sécuritaire des signaux statiques.

Un exemple de schéma de convertisseur est donné figure II.5.

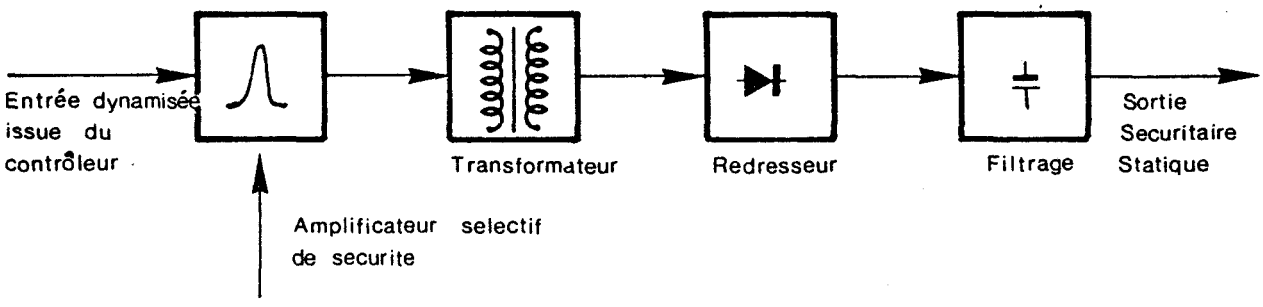


Fig II.5

II.5) DETECTION DES DEFAUTS.

On appellera défaut, l'altération, par une défaillance, d'une information localisée dans le contrôleur.

La conception en sécurité intrinsèque impose au circuit qu'il prenne l'état de sécurité défini pour lui s'il n'est plus en mesure , suite à une défaillance matérielle qui l'affecte, de remplir correctement sa fonction.

Lorsqu'une défaillance apparait, il est nécessaire d'obtenir et de vérifier que :

* soit elle est détectée et le contrôleur prend l'état de sécurité.

* soit elle n'est pas détectée (on parle alors de panne latente) mais, elle n'altère pas la sécurité ; la tâche de contrôle est toujours remplie correctement et la combinaison avec toute autre défaillance apparaissant par la suite permet également de respecter la sécurité (prise de l'état de sécurité).

On conçoit que la première situation est plus satisfaisante si le nombre de pannes susceptibles de se produire est important.

Au niveau de la démarche à adopter, il est nécessaire d'examiner deux points pour aboutir à la détection des défaillances :

- provoquer l'apparition d'un défaut mettant en évidence une défaillance (par définition matérielle).

- favoriser la propagation de ce défaut sur les circuits en aval pour que la sortie effective du contrôleur puisse prendre l'état de sécurité. Si le défaut est "absorbé" en un endroit quelconque dans le circuit, la panne est dite "latente" et donc potentiellement dangereuse.

II.5.1) MANIFESTATION D'UN DEFAUT EN LOGIQUE SECURITAIRE CLASSIQUE.

Ce type de conception est basé sur la réalisation de portes logiques sécuritaires qui, assemblées entre elles, réalisent une fonction complexe.

Chaque porte est conçue de manière à ce que toute défaillance produise un défaut "orienté", par exemple une sortie figée à l'état 0, on vérifie ceci par une démonstration de sécurité.

La modélisation des défaillances étant unique et chaque sortie de porte accessible, la conception des fonctions complexes est facilitée.

II.5.2) MANIFESTATION D'UN DEFAUT EN LOGIQUE INTEGREE.

En logique intégrée, le concepteur ne dispose que de portes normales, non sécuritaires et, de plus, non toujours directement accessibles (cas des fonctions complexes intégrées).

Lorsqu'une défaillance se manifeste, le défaut produit peut se présenter sous diverses formes (collage à 0 ou 1,etc...) sauf si le circuit est en logique alternative (Cf § IV.1).

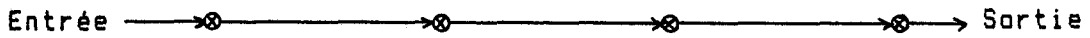
On conçoit sans difficulté que la propagation de défauts non orientés est plus problématique.

II.5.3) PROPAGATION DES DEFAUTS (REF.15).

Ce paragraphe ne prétend pas résoudre le problème de la propagation des défauts à travers une structure logique d'autant plus qu'il n'existe pas de règles générales pour les défauts non orientés, il présente simplement les problèmes rencontrés en fonction de la structure des assemblages.

Lorsqu'on représente un circuit sous forme de noeuds et de lignes (REF.15), on peut distinguer trois types de configurations :

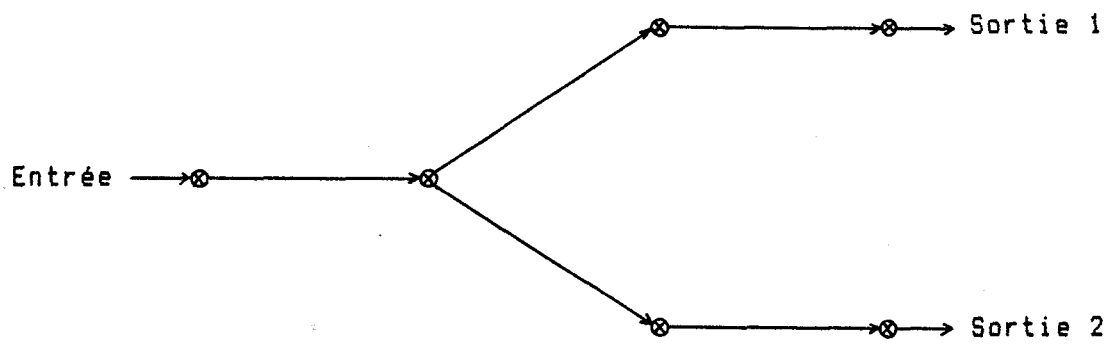
* la configuration série :



Si chaque noeud (⊗) est réceptif à un défaut produit par le noeud amont, le défaut est facilement propagé.

Le dernier noeud avant la sortie doit être conçu de manière à ne produire une sortie dynamisée que dans des conditions bien figées de ses entrées (sinon le problème n'a pas de solution).

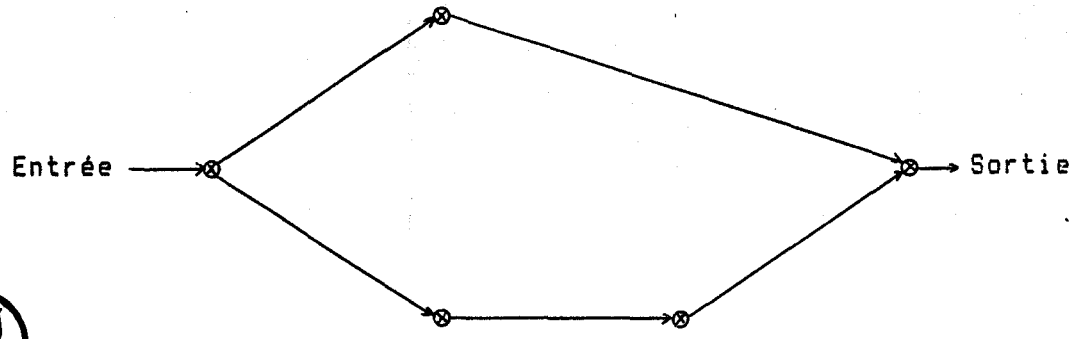
* la configuration divergente :



Lorsque le circuit dispose de sorties multiples, on trouve cette configuration.

Si le défaut apparait en amont de la divergence, il est propagé suivant plusieurs segments en série et il suffit qu'il soit détecté par une ou plusieurs sorties pour que le problème soit résolu.

* la configuration reconvergente :



Dans cette configuration, il y a d'abord divergence puis convergence : des informations élaborées par deux branches parallèles se recombinent.

Si un défaut apparait en amont de la divergence, il est propagé et transformé par les branches parallèles et à l'endroit de convergence, il est possible d'avoir une compensation: le défaut est absorbé.

Lorsque les signaux d'entrée sont détermistes, il est possible de faire une simulation en injectant tous les défauts en amont de la divergence pour constater ou non la détection.

Aucune règle de propagation ne peut être définie avec des défauts non orientés, chaque circuit avec ses signaux d'entrée est un cas particulier.

Le chapitre IV donne un exemple avec une logique à défaut orienté et un exemple avec une logique à défauts non orientés.

En tout état de cause, pour propager des défauts, il faut que ceux ci soient effectivement produits lorsqu'une défaillance apparait, c'est l'objet du chapitre suivant que d'examiner ce point de la conception.

INTRODUCTION

Ce troisième chapitre est consacré à l'influence des défaillances sur les circuits logiques et à leurs tests.

On peut rappeler que le test permet de révéler au niveau de chaque fonction logique les défaillances qui éventuellement l'affecte et de faire générer une information qui a pour finalité d'imposer au contrôleur l'état de sécurité.

L'étude est menée sur les deux grandes catégories de circuits possibles :

- les circuits combinatoires
- les circuits séquentiels

Si les premiers ne posent pas de problèmes particuliers, les seconds, par leur mode de fonctionnement spécifique, nécessitent une étude plus approfondie des mécanismes engendrés par les défaillances.

Nous examinons ainsi :

- l'influence des défaillances sur le fonctionnement
- les propriétés que doivent avoir les séquences de test pour être exhaustives.
- les conséquences d'un test incomplet

Ainsi, pour les deux catégories de circuits et dans le cadre d'un test non exhaustif, on aboutit à une probabilité d'existence de pannes latentes pouvant être contraire à la sécurité.

Cette probabilité, dont le complément à un est équivalent à un taux de couverture de pannes (τ), donne le taux résiduel d'insécurité λ_{Si} lié à l'utilisation d'une fonction logique.

$$\lambda_{Si} = \lambda_j (1 - \tau) \qquad \text{(REF 17)}$$

avec λ_j le taux de défaillance horaire du circuit.

III.1) TEST DES CIRCUITS LOGIQUES.

Le but du test est de montrer qu'un composant, en l'occurrence un circuit logique répond à son cahier des charges fonctionnel (relations entrées-sorties) et à son cahier des charges de performances (rapidité, sortance, entrance,...).

On distingue deux classes de test :

- le test hors ligne
- le test en ligne

III.1.1) TEST HORS LIGNE.

Il consiste à isoler un composant de son contexte d'utilisation et à lui appliquer un ensemble de stimulations électriques rendant compte de son bon fonctionnement.

Replacé dans son environnement d'utilisation rien n'assure (sauf faiblesses révélées par le test) que l'individu testé fonctionnera encore correctement quelques instants plus tard.

Il est évident que cette famille de test n'est pas adaptée à la réalisation de contrôleurs en sécurité intrinsèque puisque l'investigation n'est que ponctuelle.

Toutefois, il est intéressant de faire un rapide bilan sur les moyens utilisés.

Le test hors ligne peut se concevoir de deux manières :

- un test déterministe
- un test pseudo aléatoire

* Test Déterministe :

Le circuit est analysé afin d'y recenser les pannes susceptibles de se produire, on peut alors déterminer les stimulations électriques, vecteurs d'entrée ou séquences de vecteurs, qu'il faut lui appliquer pour révéler une défaillance donnée (REF.16).

* Test Pseudo aléatoire (REF.18):

C'est un test probabiliste, à partir de l'analyse du circuit (modes de défaillance), on détermine qu'elle est la panne "la plus difficile à détecter" ainsi que sa probabilité élémentaire de détection (probabilité qu'à un seul vecteur d'entrée d'une séquence pseudo aléatoire de révéler cette défaillance).

Ceci permet de calculer la longueur L minimale de la séquence nécessaire pour détecter la présence d'une panne avec une certaine probabilité de succés.

L'exemple suivant est tiré de la REF.19.

Exemple : La bascule D de type 7474 possède 4 entrées et 2 sorties et elle se compose de 6 portes NAND à 3 entrées chacune.

Pour obtenir une probabilité de détection de 0,999 ,il faut lui appliquer une séquence pseudo aléatoire de 1100 vecteurs à quatre composantes.

La défaillance que l'on cherche à révéler étant la plus difficile à détecter, on suppose que les autres par définition plus faciles à détecter, seront, elles aussi, piégées par la séquence.

Ce genre de test est intéressant lorsqu'une investigation déterministe devient excessivemenent longue.

Le test hors ligne dans le cas général dispose de moyens en matériel important :

- générateur de séquences de test
- moyens de comparaison externes tels que des circuits étalons

Ces moyens ne sont pas disponibles pour le test en ligne.

III.1.2) TEST EN LIGNE.

Il constitue, au sens de la sécurité, la seule solution satisfaisante puisque par définition, le test à lieu in situ, par le moyen des opérations normales de fonctionnement.

Durant celles ci, une certaine quantité d'informations supplémentaires, susceptibles de révéler les pannes, est ajoutée à l'information propre à traiter par la fonction. Ce supplément doit être suffisant pour garantir l'exhaustivité du test.

Le type de fonction logique, combinatoire ou séquentiel, le nombre d'entrées et de sorties sont parmi les paramètres qui déterminent la quantité de stimulations nécessaires pour faire le test en ligne.

Celui ci est limité , bien entendu, à l'aspect fonctionnel du circuit, les anomalies relatives aux performances technologiques (rapidité, niveau de tension, etc...) ne seront éventuellement révélées que si leurs dégradations entrainent des troubles sur le fonctionnel.

Par exemple :

Pour une porte, un temps de propagation qui devient prohibitif (retard à la montée d'un signal de sortie) est détecté comme une panne franche s'il provoque des aléas de commutation dans une bascule.

III.2) FONCTIONS COMBINATOIRES

Une fonction combinatoire peut être définie par la relation φ qui existe entre ses entrées et ses sorties.

A tout vecteur $\vec{V_e}$ d'entrée correspond par l'application φ un vecteur $\vec{V_s}$ de sortie qui est toujours le même.

La fonction combinatoire n'a pas, en l'absence de défaillance, la mémoire des vecteurs qui lui ont été précédemment appliqués.

$$\vec{V_e} \xrightarrow{\varphi} \vec{V_s}$$

La fonction combinatoire peut également être définie physiquement par son nombre de lignes d'entrée, noté n, et son nombre de lignes de sortie, noté s.

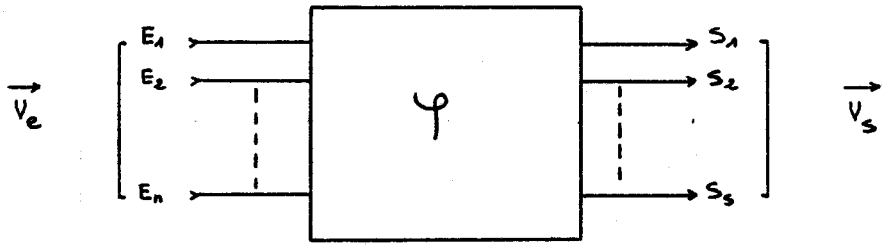


Fig.III.1

Soit $\mathcal{E}$, l'ensemble des vecteurs d'entrée possibles

Soit $\mathcal{S}$, l'ensemble des vecteurs de sortie possibles

$$\mathcal{E} \xrightarrow{\varphi} \mathcal{S}$$

En binaire :

$\mathcal{E}$ peut être dénombré par Card $(\mathcal{E}) = 2^n$

$\mathcal{S}$ peut être dénombré par Card $(\mathcal{S}) = 2^s$

La totalité des valeurs de $\mathcal{S}$ n'est pas toujours utilisée et $\mathcal{S}$ peut être partitionné en $\mathcal{S}_1$ et $\mathcal{S}_2$ tels que $\mathcal{S}_1 \cap \mathcal{S}_2 = \phi$

et $$\mathcal{E} \xrightarrow{\varphi} \mathcal{S}_1$$

L'apparition d'un vecteur de $\mathcal{S}_2$ constitue déjà la détection d'une défaillance mais ne suffit pas pour affirmer que le circuit réalisant φ est testé, il existe en particulier, des défaillances qui vont faire produire au circuit des sorties erronées mais faisant parties de $\mathcal{S}_1$.

L'application φ est représentée par une équation booléenne ou une table de vérité qui donne immédiatement la correspondance entrées, sorties (figure III.2).

Entrées				Sorties			
E_n	E_i	E_1	E_0	S_s	S_j	S_1	S_0
0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1
0	0	1	0	1	1	1	0
⋮			⋮	⋮			⋮
1	1	1	1	1	1	0	1

Fig.III.2

- Considérant le circuit comme une "boite noire", le test exhaustif ne peut se concevoir qu'en appliquant systématiquement toutes les combinaisons d'entrée possibles.

Deux cas sont alors envisageables :

1° CAS Aucune discordance entre la table de vérité obtenue et celle attendue (cahier des charges fonctionnel).

Le circuit testé remplit alors parfaitement sa fonction .

On peut supposer qu'il n'est pas le siège d'une défaillance, sauf s'il est redondant.

Remarque : Un circuit combinatoire est dit redondant s'il existe des défaillances telles que la présence de celles ci ne change rien à la fonctionnalité du circuit (REF.20). En fait, la défaillance, si elle existe, n'est pas détectable par les moyens classiques de test.

Exemple : Soit le circuit de la figure III.3 qui présente une liaison redondante ℓ.

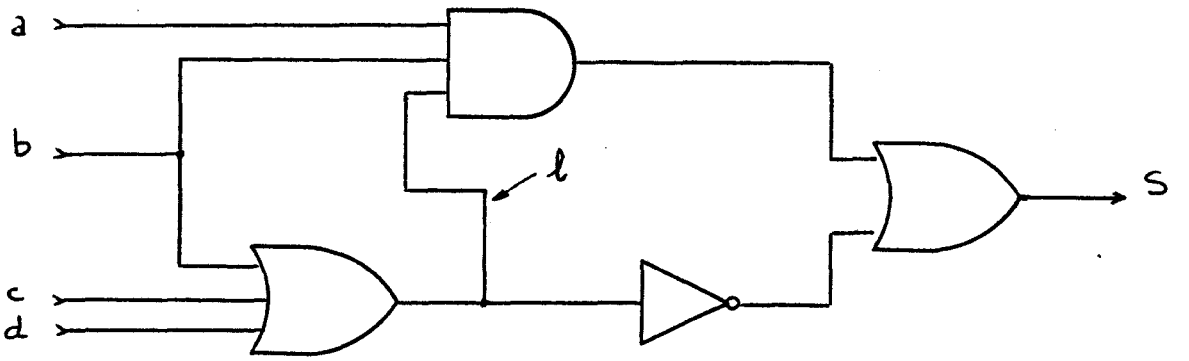


Fig.III.3 (REF.16)

L'équation est la suivante

$$S = ab(\underbrace{b+c+d}_{\ell})+\overline{bcd}$$

L'équation peut être considérée comme mal simplifiée puisque

$S = ab(b+c+d)+\overline{bcd} = ab+abc+abd+\overline{bcd} = ab(1+c+d)+\overline{bcd} = ab+\overline{bcd}$

2ᵉ CAS Une ou plusieurs divergences sont apparues.

Il y a eu production de défauts qui, si la propagation s'est correctement déroulée, font prendre au contrôleur l'état de sécurité.

Si le circuit est parfaitement connu dans sa constitution interne (niveau 2), il n'est pas nécessaire d'appliquer les 2^n vecteurs d'entrée pour être exhaustif. En effet :

- certains vecteurs sont capables de permettre la détection de plusieurs défaillances .
- la totalité des pannes recensées peut être révélée par un nombre réduit de vecteurs d'entrée.

Schéma logique interne connu

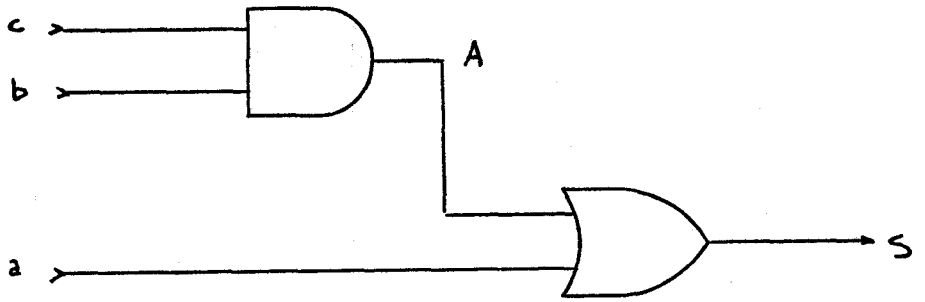


Table de vérité

c	b	a	S	
0	0	0	0	
0	0	1	1	-----> a est active seule
0	1	0	0	-----> b est inactive seule
0	1	1	1	
1	0	0	0	-----> c est inactive seule
1	0	1	1	
1	1	0	1	-----> A n'est pas collée à 0
1	1	1	1	

Cet ensemble de quatre vecteurs suffit à tester la fonction

* chaque entrée est testée pour ses deux valeurs
* la porte interne ET est testée
* la sortie S est testée

Fig.III.4

Il peut être intéressant d'évaluer la qualité du test lorsque tous les vecteurs d'entrée ne sont pas appliqués au circuit.

En effet, lorsque le circuit est très simple, il est facile de déterminer si une suite de valeurs qui lui est présentée permet de détecter toutes les défaillances modélisées. Il en va autrement si :

* le circuit n'est pas intimement connu (boite noire)
* si le circuit est trop complexe pour qu'en ligne, il soit possible d'effectuer un test exhaustif (trop de vecteurs à appliquer)

Nous allons quantifier d'une manière probabiliste la qualité d'un test non exhaustif sur un circuit à une sortie puis, sur un circuit à sorties multiples.

III.2.1) QUALITE DU TEST D'UN CIRCUIT A UNE SORTIE.

Considérons un circuit combinatoire C à n entrées et une seule sortie.

Le schéma interne est inconnu, seule la fonction Υ remplie est connue.

L'ensemble Φ représente l'ensemble des fonctions réalisables par un circuit C à n entrées et une seule sortie, Υ est un élément de cet ensemble (celui correspondant à la fonction attendue de C).

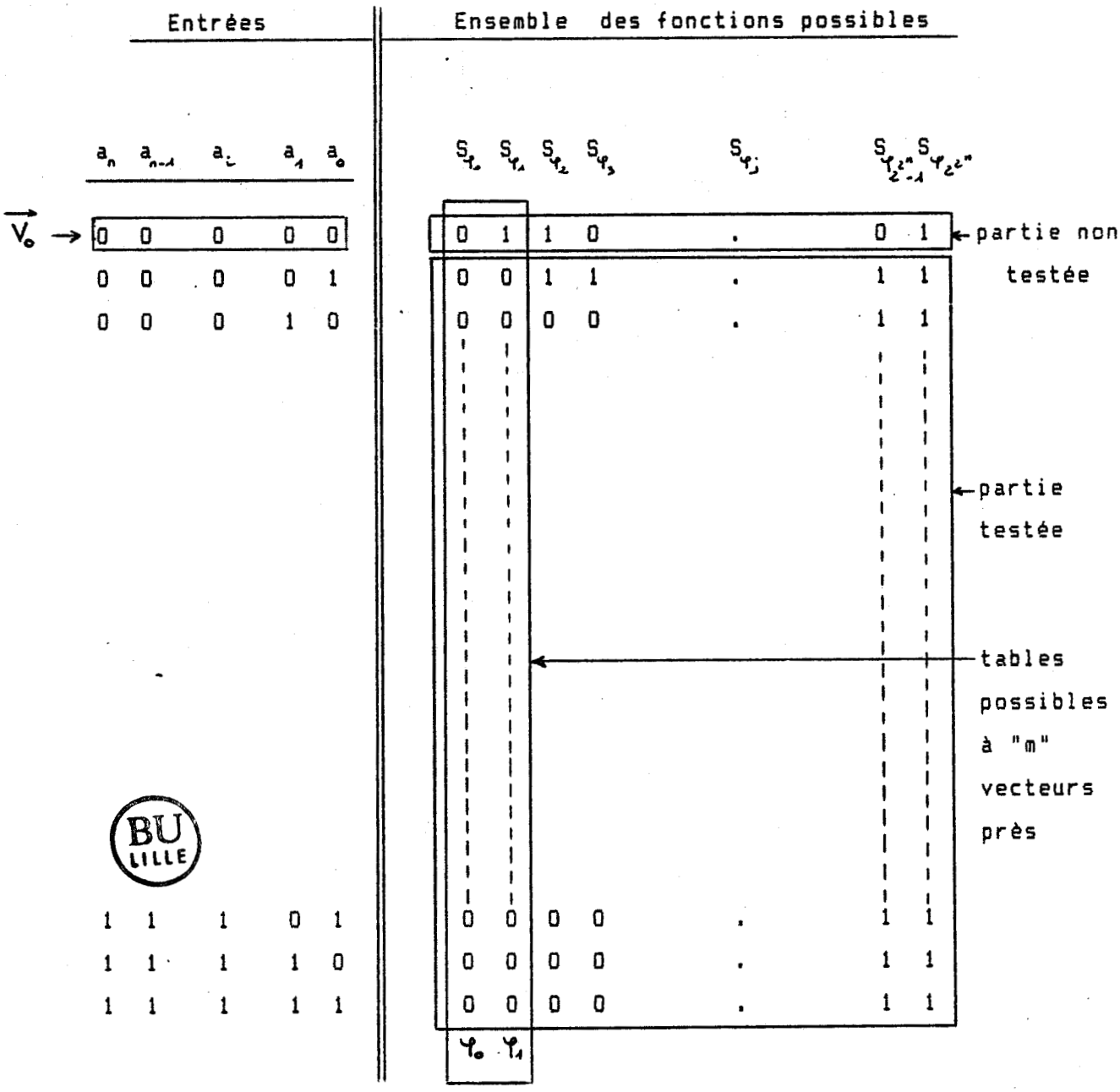
A priori, ne connaisant pas C de façon intime, on peut considérer que $\Phi - \Upsilon$ représente l'ensemble des fonctions que peut réaliser C lorsqu'il est le siège de défaillances internes.

$\text{Card}(\Phi) = 2^{2^n}$ à chaque combinaison des 2^n entrées correspond un arrangement des 2^a sorties différent d'où Card (Φ)

On suppose que le mécanisme de test est très performant et que:

$\forall \vec{V_i}$ ($\vec{V_i}$ vecteur d'entrée) $V_i \xrightarrow{\Upsilon} S_i$

si $S_{i\,reçu} \neq S_{i\,attendu}$ la panne est détectée



- la partie testée, qui correspond à la partie de la table de vérité connue puisque les vecteurs d'entrée sont appliqués donc, les sorties accessibles.

- la partie non testée correspond aux vecteurs d'entrée non appliqués, la réponse du circuit n'est pas connue pour ces valeurs.

- la troisième correspond aux tables de vérité correctes et identiques aux vecteurs d'entrée non appliqués près.

Fig.III.5

L'application de 2^n vecteurs différents de $\mathcal{E}$ valide le circuit C avec une certitude, le test est alors exhaustif.

La probabilité que C remplisse la fonction Ψ vaut 1

$$\text{Prob}_{2^n}(C = \Psi) = 1$$

Au contraire, si aucun vecteur n'est soumis au circuit, l'observateur se fie entièrement au hasard et il possède 1 chance sur 2^{2^n} d'avoir le circuit qu'il souhaite ou d'avoir son circuit C exempt de toute défaillance, car il n'existe qu'un seul circuit de l'ensemble qui satisfasse exactement à la table de vérité de Ψ.

$$\text{Prob}_{0}(C = \Psi) = \frac{1}{2^{2^n}}$$

Entre ces deux cas extrêmes, il existe tout un ensemble de tests tronqués dont le paramètre de différentiation peut être le nombre "m" de vecteurs non appliqués au circuit considéré.

Sans autres formalités et par souci de simplicité prenons:

- m=1, le vecteur correspondant est V_o= (0,0,0, ...,0,0)
- $\Psi = \Psi_o$, cette fonction correspond à la fonction NON

La figure III.5 représente l'ensemble des tables de vérité possibles pour C.

Après avoir éliminé toutes les parties que le test permet de discriminer, on constate qu'avec les paramètres que nous nous sommes fixés, il ne subsiste que deux tables qui ne se différencient que pour une valeur d'entrée correspondant au vecteur V_o exclu du test.

et donc:

$$\text{Prob}_{2^n-1}(C = \Psi_o) = P_{m=1}(\Psi_o) = \frac{2^{2^n} - 2 + 1}{2^{2^n}}$$

pour m=2 , le nombre de tables non entièrement testées et susceptibles d'être correctes vaut $2^2 = 4$; en généralisant :

$$P_m(\Psi_o) = 1 - \frac{2^m - 1}{2^{2^n}}$$

$P_m(\Psi_o)$ est égal, en fait au rapport du nombre de circuits défaillants détectés au nombre de circuits possibles. En ce sens, elle peut être assimilée au taux de couverture du test, c'est à dire:

$$P_m(\Psi_o) = \frac{\text{Nombre de circuits défaillants détectés}}{\text{Nombre de circuits possibles}} = \tau$$

Le complément à 1 de $P_m(\Psi_o)$ représente la probabilité que le test ne révèle pas une défaillance du circuit C

$$Q_m(\Psi_o) = 1 - P_m(\Psi_o) = \frac{2^m - 1}{2^{2^n}} = 1 - \tau$$

III.2.2) QUALITE DU TEST POUR UN CIRCUIT A s SORTIES.

Les circuits concernés possèdent n entrées et plusieurs sorties (s) sur lesquelles on peut vérifier la présence de défauts.

Card $(\mathcal{E}) = 2^n$ qui concerne l'entrée n'a pas lieu de changer.

Card $(\Phi) = 2^{2^n} . 2^{2^n.(s-1)}$

En effet, à chaque table de vérité précédente, s'ajoute un nombre de tables équivalent aux combinaisons des s moins une autres tables issues des s moins une autres sorties supplémentaires. Il en est de même pour le nombre de tables non testées lorsque m vecteurs sont exclus du test exhaustif, le nombre de ces tables non testées est multiplié par 2^s.

Il vient alors que :

$$P_{sm}(\Psi_o) = 1 - \frac{2^{ms} - 1}{2^{2^n + 2^n(s-1)}} = \tau$$

et

$$Q_{sm}(\Psi_o) = \frac{2^{ms} - 1}{2^{2^n + 2^n(s-1)}} = 1 - \tau$$

III.2.3) SYNTHESE.

Tester une fonction combinatoire consiste à lui appliquer un grand nombre de vecteurs différents en entrée, en fait 2^n pour être exhaustif.

S'il existe dans le circuit réalisant cette fonction une défaillance, celle ci est révélée par un ou plusieurs vecteurs de sortie faux.

Dans le cas d'une approche de type "boite noire", l'inconvénient réside dans le fait que les 2^n entrées doivent être appliquées si l'on veut que l'approche soit parfaitement déterministe donc en accord avec le concept de sécurité intrinsèque que nous essayons de conserver.

La connaissance des défaillances internes possibles et des schémas permet de réduire la longueur du test ($m \neq 0$) tout en restant parfaitement déterministe.

Il nous a semblé intéressant d'évaluer le degré de connaissance qu'on a d'un circuit combinatoire lorsque la séquence de test qui lui est appliquée n'est pas exhaustive.

Le calcul permet d'obtenir une probabilité qui reflète la possibilité qu'a un certain nombre de pannes de ne pas être vu et d'être ainsi des pannes latentes que par excès nous considérons comme potentiellement dangereuses.

On en déduit ainsi le taux d'insécurité résiduel, λ_{si}, résultant de l'emploi de ce circuit combinatoire insuffisamment testé dans le cadre du contrôleur de sécurité.

$$\lambda_{si}(C) = (1 - \tau) . \lambda_s(C) = Q_{sm}(C) . \lambda_s(C)$$

Si $Q_{sm}(C) = 0$, le circuit est totalement testé et aucune insécurité résiduelle n'existe, nous sommes en sécurité intrinsèque.

Au contraire, si $Q_{sm}(C) = 1$, la sécurité est purement fiabiliste.

III.3) LES FONCTIONS SEQUENTIELLES.

Une machine qui réalise une fonction F séquentielle possède une faculté de mémorisation de ses états antérieurs. La sortie dépend non seulement de l'entrée qui lui est appliquée à cet instant mais aussi des entrées qui lui ont été appliquées auparavant.

On distingue deux types de fonctionnement principaux :

- le mode synchrone
- le mode asynchrone

Dans le mode synchrone, le basculement des sorties, changement de valeur, s'effectue en synchronisme avec un signal d'horloge, sur une transition de celui ci. Hors transitions, l'état de la sortie n'est nullement affecté par l'ensemble des variations susceptibles de se produire en entrée.

L'horloge n'est pas considérée comme une entrée et n'apparait pas dans l'analyse fonctionnelle de la machine.

En mode asynchrone, chaque changement à l'entrée peut modifier la sortie; l'entrée d'horloge, lorsqu'elle existe, est banalisée.

On peut signaler que l'analyse d'un circuit fonctionnant selon le premier mode peut être faite en mode asynchrone en redonnant à l'entrée particulière qu'est l'horloge un rôle d'entrée normale.

En ce qui nous concerne et pour l'analyse fine de la fonction en cas de panne, il est intéressant d'avoir une approche de type asynchrone .

Avec celle ci, le circuit est décrit de manière plus intime, en particulier on voit apparaitre toutes les composantes du vecteur d'entrée, y compris l'horloge, et leurs rôles respectifs dans l'évolution interne de la machine. En contrepartie, les représentations graphiques sont plus volumineuses.

III.3.1) REPRESENTATION EN MODE ASYNCHRONE.

Nous allons adopter tout au long de ce rapport la représentation suivante pour les fonctions séquentielles.

Soit, nous aurons une table d'évolution dont un exemple commenté est donné figure III.6, soit la fonction sera représentée par un graphe; la figure III.7 est le graphe associé à la table précédente.

Table d'évolution d'une machine à 2 entrées, 2 sorties, 6 états

Entrées / Etats	00	01	10	11
S_1 0111	(S_1),11	S_2	S_5	
S_2 1110	S_1	(S_2),11		S_3
S_3 1010		S_2	S_4	(S_3),01
S_4 1011	S_1		(S_4),01	S_3
S_5 0101	S_1		(S_5),10	S_6
S_6 0101		S_2	S_5	(S_6),10

ABCD

← Vecteurs d'entrée.

← Si la machine se trouve dans l'état S_1 suite à l'entrée 00, il ne sera pas possible d'arriver dans cette case directement: la simultanéité du changement de 2 entrées n'est pas admise.

Nom des états de la machine suivi du code binaire associé

Etat de la machine suivi du vecteur de sortie associé.

Les états non cerclés correspondent à des états transitoires qui précèdent l'état effectif atteint.

Fig .III.6

Graphe associé

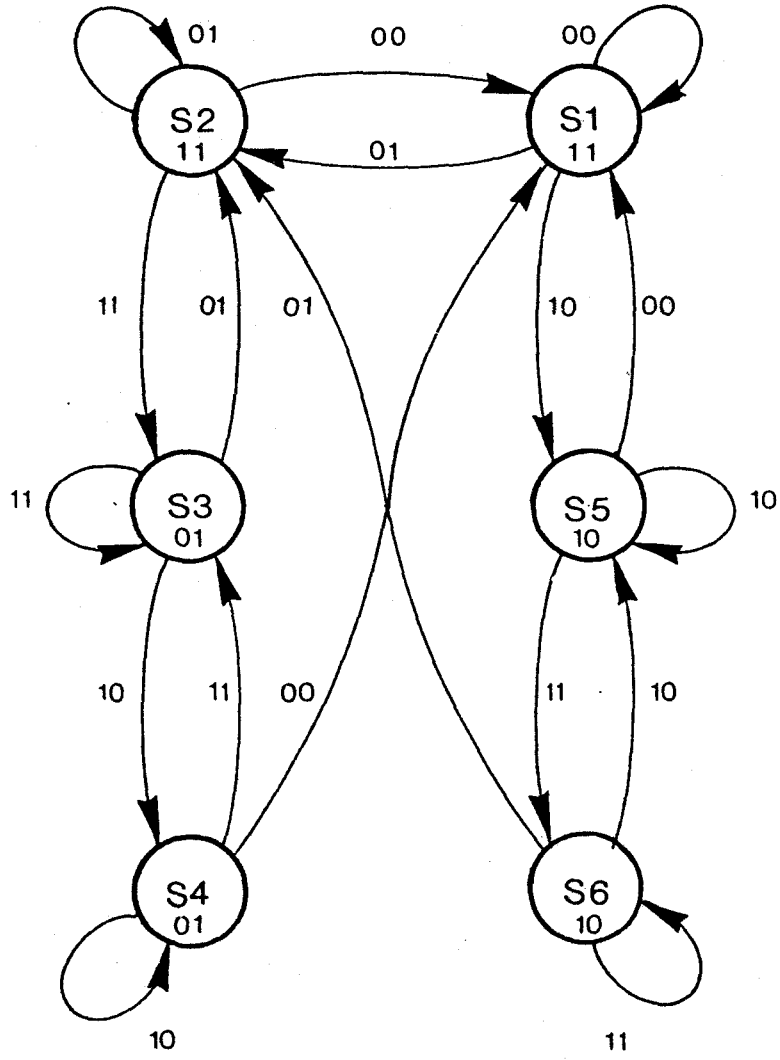


Fig.III.7

A partir de la représentation du fonctionnement en mode asynchrone on déduit les éléments suivants :

* une seule entrée à la fois peut changer de valeur, ceci correspond à un arc reliant deux états sur le graphe

* à chaque état est associé un seul vecteur de sortie

* de chaque état partent n+1 arcs dont un qui permet de rester dans le même état (nous ne le représenterons plus par la suite)

* le nombre total d'arcs d'une machine, en ne comptant pas les arcs partant et arrivant dans le même état, vaut

$A = S.n$ avec S le nombre d'états de la machine

et n le nombre d'entrées de la machine

III.3.2) INCIDENCE D'UNE DEFAILLANCE SUR LE GRAPHE.

La machine étant représentée par un graphe, il peut être intéressant d'examiner les modifications induites par des défaillances sur celui ci.

A priori, le graphe ne présume pas de la façon dont est réalisé technologiquement le circuit, l'analyse des modifications de ce graphe est donc au départ très générale.

La fonction peut se représenter comme une machine de Moore (Fig.III.8) avec :

- une machine d'états δ réalisant l'enchainement des états en fonction de l'entrée et de l'état précédent.
- une fonction combinatoire γ de sortie qui à partir des états générés réalise la sortie souhaitée pour chaque état.

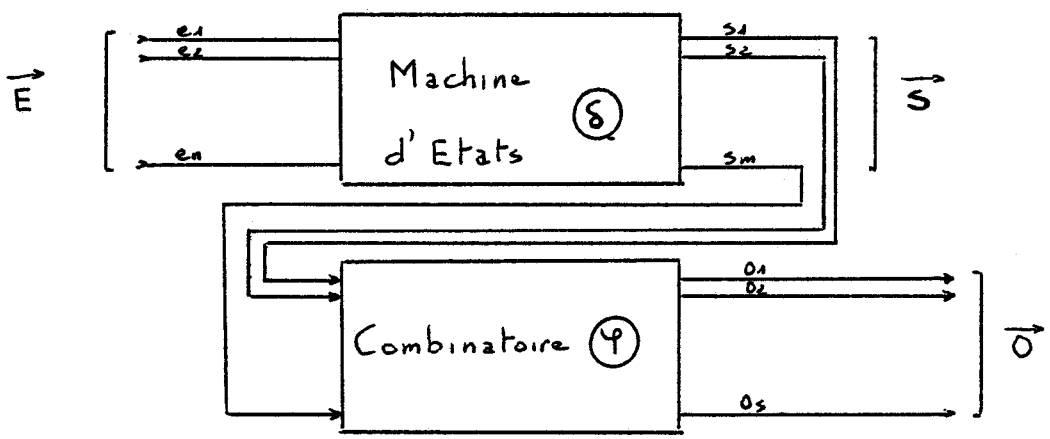


Fig III.8

On peut trouver les modifications possibles en examinant les défaillances situées sur ces deux parties. Nous aurons ainsi :

* les modifications de la fonction de sortie qui vont changer la valeur du vecteur de sortie pour un ou plusieurs états

* les modifications de la machine d'états qui vont provoquer des changements dans la topographie.

Bien entendu, c'est deux types d'altérations, suivant les défaillances, peuvent être présentent simultanément.

III.3.3) MODIFICATIONS DE LA VALEUR ASSOCIEE AUX ETATS.

Des défaillances comme celles modélisées sur les portes logiques du chapitre II provoquent ce genre de défauts :

- rupture de la liaison entre machine d'états et combinatoire γ
- coupure de liaison entre le combinatoire et la sortie effective
- modification de la fonction γ due à une défaillance sur une porte la réalisant

Dans la machine donnée en illustration aux figures III.6 et 7, le combinatoire est réduit puisque ce sont les sorties s_2 et s_3 qui donnent le vecteur de sortie de chaque état.

Nantie de cette défaillance, la machine continue à avoir un comportement normal mais les sorties qu'elle délivre sont fausses.

III.3.4) MODIFICATIONS DE LA TOPOGRAPHIE.

Cette modification affecte la machine d'états et on peut imaginer plusieurs conséquences sur le graphe :

- l'apparition d'états nouveaux; la machine a plus d'états qu'avant la défaillance

- la modification de la destination des arcs; l'enchainement logique des états ,qui singularisait la machine, n'est plus le même.

On peut examiner les détails de ces comportements puisque ceci est nouveau par rapport aux fonctions combinatoires.

III.3.4.1) CREATION D'ETATS

Soit $\mathcal{S}$, l'ensemble de tous les états possibles avec des vecteurs d'état à s composantes

En binaire, Card $(\mathcal{S}) = 2^s$

$\mathcal{S}$ peut être partitionné en $\mathcal{S}_1$ et $\mathcal{S}_2$ tels que $\mathcal{S}_1$ constitue l'ensemble des états effectivement utilisés par construction en l'absence de pannes, $\mathcal{S}_2 \neq \emptyset$.

On soupçonne donc que suite à une défaillance ,des états appartenant à $\mathcal{S}_2$ vont apparaitre.

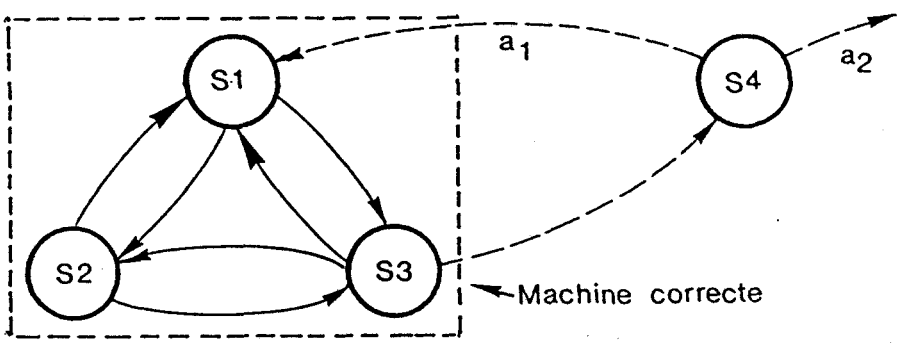


Fig III.9

On peut considérer que ces états nouveaux existent dans le graphe mais ne sont pas accessibles car, par construction, il n'y a pas d'arcs les reliant aux états normaux $(\mathcal{S}_1)$ de la machine correcte.

La figure III.9 donne l'exemple d'une machine ayant une défaillance telle qu'un état S_4 apparait en devenant accessible.

L'apparition n'est possible que si un arc de la machine correcte change de destination, on remarquera que le nombre d'arcs de la nouvelle machine obtenue augmente car à partir de S_4 ,il existe n arcs qui partent soit vers de nouveaux états créés, soit vers la machine initiale. Le nombre d'arcs ainsi créés est égal au nombre d'états rendus accessibles multiplié par le nombre d'entrées de la machine.

III.3.4.2) MODIFICATION DE LA DESTINATION DES ARCS

Un arc change de destination ,nous avons vu un cas semblable au paragraphe précédent, mais il n'y a pas ici de création d'états.

La machine possède toujours les états définis sur le circuit correct.

Ceci peut d'ailleurs conduire à des situations extrêmes telles que celles décrites à la figure III.10

- Figure III.10.a : substitution d'un état en un état bouchon, les arcs modifiés ne permettent plus de s'extraire de l'état S_3.

- Figure III.10.b : un état est rendu inaccessible car aucun arc n'y aboutit plus.

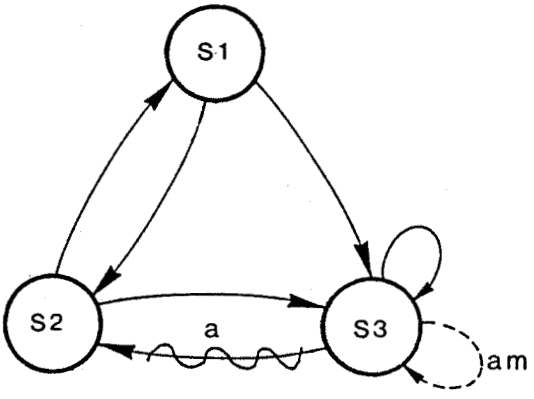


Fig.III.10.a

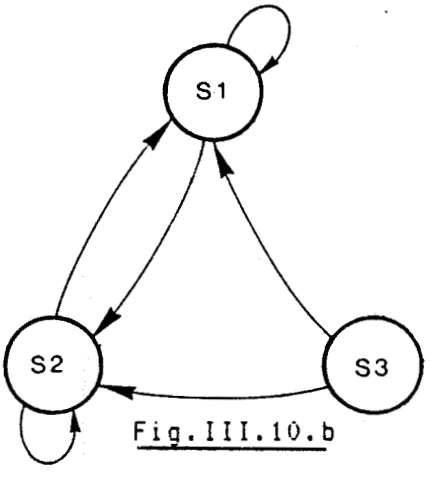


Fig.III.10.b

III.4) TEST D'UNE MACHINE SEQUENTIELLE.

Le test ne peut plus être mené , comme en combinatoire, en appliquant simplement tous les vecteurs d'entrée une seule fois; en séquentiel, selon l'état précédent, la réponse à une même entrée peut être différente.

Il est nécessaire de connaitre l'état de la machine à un instant donné, certaines séquences dites séquences de synchronisation placent invariablement la machine dans un état déterminé avant de pouvoir appliquer le test de façon significative. Cet état constitue alors, un repère à partir duquel la séquence de sortie est coordonnée à une séquence d'entrée.

Une séquence suffisante de test parcourt non seulement tous les arcs, donc passe par tous les états, mais emprunte des chemins permettant de lever toute ambiguité sur certaines modifications non immédiatement détectables.

Il est intéressant d'examiner qu'elle doivent être les propriétés des séquences pour détecter les pannes modélisées sur le graphe.

III.4.1) TEST D'UNE MACHINE SANS CREATION D'ETAT.

Les pannes entrainent des changements de deux ordres

* une modification de la fonction de sortie

La détection est ici immédiate si la séquence de test passe par tous les états du graphe, on s'aperçoit que la sortie n'est pas conforme à ce qui était attendu.

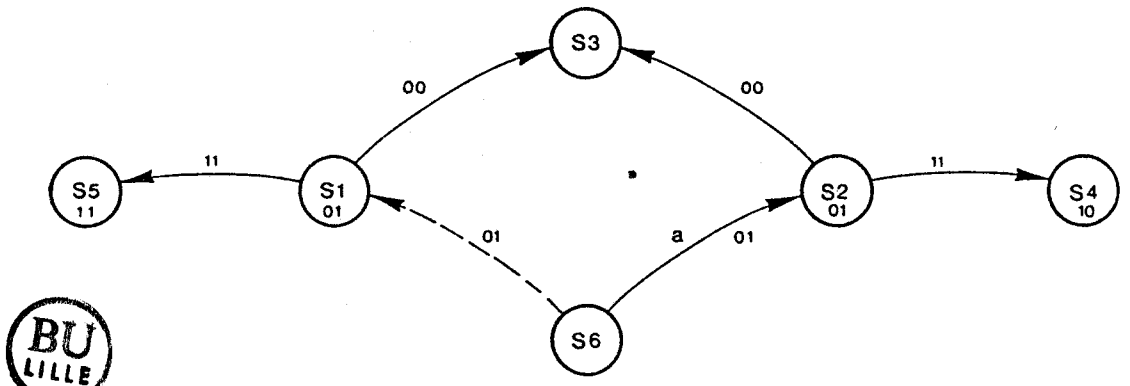
C'est en fait un simple test de type combinatoire consistant à balayer non pas tous les vecteurs d'entrée mais tous les états de la machine.

* une modification de la destination des arcs

La difficulté de cette modification a pour origine le fait que plusieurs états distincts font délivrer à la machine une même sortie. Dans ces conditions, savoir si l'état atteint est le bon ou, si suite à une défaillance, l'arc parcouru a changé de destination et a atteint un autre état possèdant une sortie identique n'est pas immédiat.

Connaissant le graphe, il est possible de trouver pour chaque défaut modélisé ,une séquence permettant de lever l'ambiguité.

La figure III.11 donne un exemple de cas d'indécision et la séquence de test permettant de lever le doute.



L'arc "a" reliant S_6 à S_2 est dérouté, il relie après l'apparition d'une défaillance S_6 à S_1.

On constate que S_1 , S_2 possèdent une sortie identique.

La séquence, au départ de S_6, consistant à appliquer au circuit les vecteurs d'entrée suivants: 01,00 aboutit à l'état S_3 avec ou sans défaillance et, la panne n'est pas détectée.

Par contre , toujours au départ de S_6, la séquence :01, 11 permet de lever l'ambiguité dans la mesure où S_5 est atteint au lieu de S_4 lorsqu'il n'y a pas cette défaillance.

Fig III.11

III.4.2) TEST D'UNE MACHINE AVEC CREATION D'ETATS.

La création d'états requiert à priori la modification de la destination d'un ou de plusieurs arcs, nous allons toutefois examiner les conditions de création d'états sans altération d'arcs de la machine correcte.

* modification avec changement dans les arcs

Le problème a été partiellement résolu dans le paragraphe précédent, il suffit que la séquence de test passe par tous les états pour que l'on constate que des états non attendus produisent une sortie non attendue.

S'il existe une fonction combinatoire de sortie (Υ), celle ci peut être construite, si le circuit n'est pas intégré, de manière à rejeter les états nouveaux en produisant un vecteur de sortie particulier. Ceci constitue une détection naturelle des états créés par une panne : Circuits combinatoires à codes disjoints (REF.21)

En cas d'impossibilité, il faut étudier en détail les nouveaux graphes pour déterminer des séquences de test susceptibles de mettre en évidence ces états perturbateurs.

* modification sans changement dans les arcs originels

Puisqu'il y a possibilité de création d'états supplémentaires, on peut s'interroger sur l'existence de machines parallèles à la machine correcte.

Ces machines ne peuvent exister que si $\mathcal{S}_2 \neq \emptyset$

La machine est en fait composée de 2 états parmi lesquels il existe Card ($\mathcal{S}_1$) états "corrects" et Card ($\mathcal{S}_2$) états qui peuvent avoir des liaisons entre eux et constituer des machines à part entière mais qui par construction ne sont pas accessibles.

Entre la machine correct et ces machines parallèles les liens sont unidirectionnels :

Les arcs partent de la machine parallèle vers la machine correcte uniquement (Figure III.12).

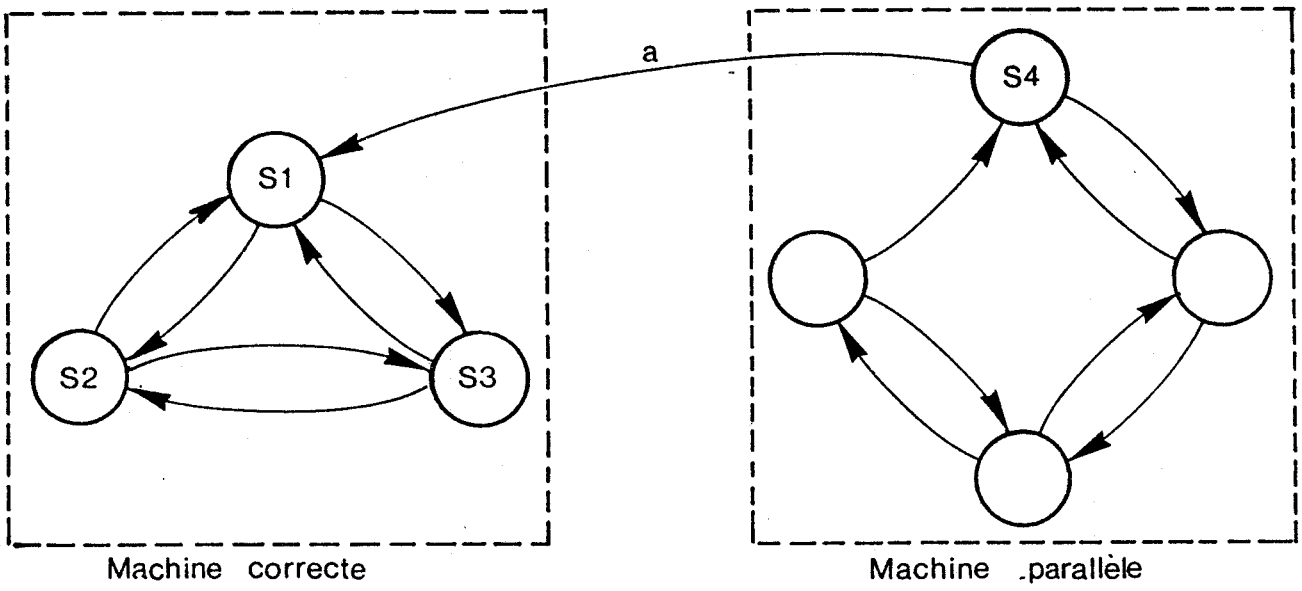


Fig.III.12

En imaginant qu'un évènement fasse passer subitement le point de fonctionnement de la machine correcte vers la ou une des machines parallèles, le comportement du circuit peut être dangereux.

Les passages dans ces graphes parallèles sont dûs:

- à un parasite électromagnétique qui à un instant provoque l'apparition dans la machine d'états (généralement composée de bascules rebouclées entre elles) d'un état appartenant à $\mathcal{S}_2$.

- à un aléa de fonctionnement , tel le retard à la commutation d'une bascule, créant ainsi la présence d'un état indésirable.

- à une panne fugitive

- à un parasite véhiculé par les alimentations électriques.

La détermination de ces graphes parallèles s'obtient par une connaissance suffisante du circuit de la machine, schéma fonctionnel composé des portes logiques élémentaires. Il suffit alors, par une analyse de ce schéma :

- d'imposer en sortie de la machine d'états, les états éléments de $\mathcal{S}_2$ et d'observer l'évolution du graphe et l'enchainement des états.

- de simuler des retards sur les portes élémentaires.

Le test pour ce genre de modification de graphe est identique à celui du cas précédent. La difficulté provenant essentiellement de la détermination des machines parallèles et de leurs graphes.

III.4.3) CONCLUSION SUR LE TEST.

Le test exhaustif d'une machine séquentielle semble être une entreprise considérable.

En examinant sur le graphe le nombre de modifications qu'il est possible de trouver, on conçoit que le nombre de séquences de vecteurs de test est nécessairement très important. Dans ces conditions, le test en ligne déterministe et complet ne peut probablement pas être mené à terme.

On peut bien entendu tempérer ce pessimisme à la lumière d'une étude réelle des modes de défaillances susceptibles d'effectivement se produire, il est également probable qu'une séquence donnée permet de détecter plusieurs types de modification, toutefois et malgré toutes ces tentatives de réduction, les séquences risquent encore d'être trop longues pour un test en ligne.

Ainsi, et de même que pour les fonctions combinatoires, nous allons quantifier la qualité d'un test partiel en faisant une approche de type probabiliste de celui ci et, de même , nous aboutirons à un taux résiduel d'insécurité résultant de l'utilisation d'une fonction séquentielle dans un circuit de sécurité.

III.5) TEST PARTIEL D'UN CIRCUIT SEQUENTIEL.

L'approche que nous allons adopter est ici quelque peu différente de l'approche utilisée pour les circuits combinatoires, la nature des fonctions séquentielles ne se prétant pas facilement à ce qui a été déjà fait.

III.5.1) LIMITE INFERIEURE DE LA PROBABILITE DE DETECTION.

Les paragraphes précédents ont montré la grande diversité des modifications possibles d'un graphe. Ces modifications élémentaires peuvent d'ailleurs, à l'occasion d'une défaillance particulière, s'associer entre elles (modifications de sorties et changements de destination des arcs).

Une séquence de test en ligne de longueur limitée aura dans ces conditions plus ou moins de "chance" de détecter une modification et, en particulier certaines défaillances auront plus de difficulté à être détectées.

Pour aboutir à un calcul très général, une approche identique à celle utilisée pour le test hors ligne à séquence pseudo aléatoire semble adaptée à notre problème (REF.19).

En effet, si l'on évalue la probabilité de détection (ou de non détection) de la panne la plus difficile à détecter, il est normal de penser que celle ci constitue un minorant de la probabilité de détection globale de la séquence de test choisie et donc, qu'avec cette même séquence, les autres défaillances, par définition plus faciles à détecter, auront une probabilité de détection supérieure.

C'est cette démarche que nous allons adopter dans la suite de ce chapitre.

Comme nous avons un modèle de la machine que nous étudions, c'est sur celui ci que nous allons rechercher la modification de graphe la plus difficilement détectable.

III.5.2) DETERMINATION DE LA MODIFICATION LA PLUS DIFFICILE.

La modification la plus difficile à détecter est intuitivement celle qui conserve un graphe très proche de celui de la machine correcte. La défaillance associée ne produit que quelques changements de destination d'arcs sur le modèle.

Pour justifier ce choix, nous reprenons les modifications modélisées.

* modification des sorties :

Celle ci est aisément détectée puisque en sortie apparaissent des vecteurs non attendus à l'instant d'observation et, intuitivement, on imagine que cette modification n'est pas très difficile à appréhender.

* création d'états :

La machine évolue parmi des états qui n'ont pas le même enchainement logique que celui de la machine correcte en produisant de plus des sorties non attendues. Nous ferons toutefois le calcul avec ce type de modification.

* suppression d'états :

Le nombre d'états étant réduit, il y a génération d'un nombre limité de vecteurs de sortie.

* modification des destinations des arcs :

La machine possède alors une topographie sensiblement identique à celle de la machine non défaillante, on retrouve un enchainement logique proche de ce qui est attendu avec des sorties correctes. La probabilité de non détection semble à priori plus importante.

III.5.3) HYPOTHESES ET DEMARCHE DU CALCUL.

* hypothèses du calcul

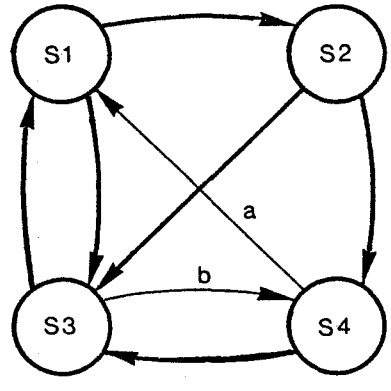
1) Partant du fait que le test qui est appliqué s'avère nécessairement trop modeste pour détecter toutes les défaillances, nous ne prendrons pas une séquence donnée, donc déterministe, mais une séquence aléatoirement distribuée sur le graphe. Cette séquence a pour caractéristiques :

- une longueur finie
- elle ne passe pas par tous les arcs du graphe de la machine correcte, nous appellerons "m" le nombre d'arcs non parcourus.

2) La défaillance possède un caractère aléatoire dans la mesure où elle est uniformément distribuée sur le graphe. Elle est caractérisée par le nombre d'arcs qu'elle affecte en changeant leur destination, nous appellerons "a" le nombre d'arcs affectés.

* démarche de calcul

Dans un premier temps et en admettant la présence de la défaillance, on calcule la probabilité que celle ci a d'être localisée sur les arcs effectivement parcourus par la séquence de test. Les arcs parcourus constituent la partie testée de la machine (Figure III.13).



les arcs en traits forts sont les arcs parcourus par la séquence de test, les arcs 1 et 2 ne sont pas parcourus, s'ils sont l'objet de la modification ,la panne ne peut pas être détectée.

Fig.III.13

Dans un second temps et considérant que la défaillance est localisée dans la partie testée (en tout ou en partie), nous calculons la probabilité pour que celle ci soit détectée par la génération d'une sortie non attendue.

Le premier calcul donne la probabilité de localisation
Le second calcul donne la probabilité de propagation

La probabilité de détection est constituée par le produit des deux car, en effet, il faut qu'il y ait localisation et propagation : l'ensemble produisant une sortie erronée.

III.5.4) PROBABILITE DE LOCALISATION.

Nous allons examiner les deux cas possibles, c'est à dire sans et avec création d'états supplémentaires.

III.5.4.1) SANS CREATION D'ETAT.

Soit une machine séquentielle telle que :

S = le nombre d'états de la machine
n = le nombre d'entrées de la machine
A = S.n = le nombre d'arcs de cette machine
a = le nombre d'arcs affectés par une défaillance

Nous allons déterminer le nombre $\mathcal{A}$ de machines différentes qu'il est possible de trouver ayant "a" arcs modifiés puis, le nombre $\mathcal{M}$ de machines ayant des arcs modifiés mais tels que cette modification ne soit pas détectable .

Le rapport $\frac{\mathcal{A}-\mathcal{M}}{\mathcal{A}}$ représente la probabilité de localisation que nous recherchons.

Soit une défaillance particulière telle qu'un arc x est altéré.

x peut prendre S directions différentes et N = S représente le nombre de machines possibles avec un arc x différent à chaque fois.

Si x et y sont altérés $N = S^2$

Avec a arcs particuliers $N = S^a$

Si on généralise en banalisant les arcs affectés de façon que "a" arcs quelconques soient affectés par une défaillance, le nombre de machines possibles différentes ayant "a" arcs qui changent de direction est égal au produit de N par les combinaisons de "a" arcs parmi les A que la machine correcte possède.

D'où :

$\mathcal{A} = S^a . \int_A^a$ = nombre de machines à "a" arcs modifiés

Le test omet de parcourir m arcs parmi A et le nombre $\mathcal{M}$ de machines pour lequel la défaillance se situe sur les arcs non parcourus vaut:

$\mathcal{M} = S^a . \int_m^a$ = nombre de machines à pannes non détectables

D'où la probabilité que la panne affecte la partie testée :

$$P_{a,m} = \frac{\mathcal{A} - \mathcal{M}}{\mathcal{A}} = 1 - \frac{S^a \int_m^a}{S^a \int_A^a} = 1 - \frac{m!}{A!} \cdot \frac{(A-a)!}{(m-a)!} \qquad \text{avec } m \geqslant a$$

Remarque : Si $m < a$, il y a plus d'arcs affectés que d'arcs non parcourus ce qui signifie que la défaillance est détectable parce qu'au moins un arc défaillant fait partie des arcs parcourus par le test.

Donc si $m < a$, $P_{a,m} = 1$

On peut définir également le complément à 1 de $P_{a,m}$ noté par $Q_{a,m}$ et qui représente la probabilité qu'une défaillance ne soit pas localisable par la séquence de test.

Le tableau suivant donne les valeurs de $Q_{a,m}$ paramétrées en a et m pour une machine ayant des caractéristiques telles que:

S = 8 états n = 2 entrées

a → / m ↓	1	2	3	4	5	10
1	0,062	0,12	0,19	0,25	0,31	0,62
2	0	$8,3.10^{-3}$	0,025	0,05	0,08	0,38
3	0	0	$1,8.10^{-3}$	$7,1.10^{-3}$	0,018	0,21
4	0	0	0	$5,5.10^{-4}$	$2,7.10^{-3}$	0,11
5	0	0	0	0	$2,3.10^{-4}$	0,06

Commentaire: La probabilité de non localisation est décroissante en fonction du nombre d'arcs affectés, en effet plus "a" est grand et plus la défaillance a de chance d'affecter la partie testée.

Au contraire, elle est croissante en fonction de m puisque cette fois la partie testée de la machine diminue.

Exploitation: Ce résultat est valable quelle que soit la séquence de test, sachant que celle ci omet de parcourir m arcs du graphe.

Pour utiliser pratiquement l'expression de $P_{a,m}$, il suffit d'analyser le circuit pour trouver le nombre "a" , ceci se fait en injectant les défaillances modélisées sur les portes et en recherchant le graphe nouveau correspondant, par comparaison on obtient "a".

En particulier, la probabilité de localisation la plus mauvaise est obtenue pour la panne donnant une valeur de "a" la plus petite.

III.5.4.2) AVEC CREATION D'ETATS.

La création de nouveaux états s'accompagne nécessairement du changement d'un ou plusieurs arcs de la machine correcte sauf, dans le cas des machines parallèles que nous traiterons à part.

Avec modification des arcs, le calcul est le même puisque :

- soit les arcs déviés vers ces états appartiennent à la partie testée de la machine et la localisation est possible.

- soit les arcs déviés ne sont normalement pas parcourus et il n'y a pas localisation .

ces conditions sont les mêmes qu'au paragraphe précédent.

probabilité de localisation dans le cas de machines parallèles.

En ce qui concerne les machines parallèles, la démarche est différente car il ne peut y avoir à priori, de localisation sur ce qui n'existe pas (pas d'arc dévié, le passage peut être considéré comme une déviation fugitive d'un arc donc, indétectable).

Ceci ne signifie pas que l'évolution dans un graphe parallèle puisse passer inaperçue, en effet, l'enchainement des nouveaux états n'est pas semblable à celui de la machine originelle. La défaillance est détectable par génération de sorties non attendues et on peut effectuer le calcul avec la probabilité de propagation du paragraphe suivant.

Dans ce cas particulier, on prend, pour cette défaillance si elle est possible, une probabilité de localisation égale à 1, ce qui semble être logique puisque c'est la machine toute entière qui est modifiée .

III.5.4.3) CONCLUSION

La probabilité de localisation permet de chiffrer ce que l'on est capable de pouvoir détecter.

La défaillance est en effet potentiellement détectable car elle affecte une partie utilisée de la machine

Partant de ce constat, les machines qui seront employées devront être optimisées pour la fonction à remplir et ne pas comporter en particulier d'entrées inutiles puisque celles ci augmentent nécessairement le nombre d'arcs d'où une surcharge de test.

Ainsi une bascule D de type 7474 possède :

24 états et 96 arcs car elle dispose de deux entrées supplémentaires ,remise à 0 et remise à 1.

Si ces entrées ne sont pas utilisées, le volume de test devient très important.

Une bascule D optimisée en supprimant ces deux entrées possède:

8 états et 16 arcs (Annexe 2).

Une approche de type boite noire qui peut sembler ne pas permettre la recherche de la défaillance la plus difficile est encore possible, il suffit de prendre pour celle ci une valeur de "a" égale à 1 et, il faut connaitre le graphe de la machine mais ceci fait partie du minimum indispensable.

$P_{s,m}$ ne constitue qu'une probabilité potentielle de détection, lorsque le défaut est détectable, il faut encore parvenir à le faire sortir de la machine sous la forme d'un défaut.

III.5.5) PROBABILITE DE PROPAGATION.

On considère cette fois qu'il existe une défaillance et que celle ci est localisée dans la partie testée de la machine.

Parce qu'il existe des états délivrant des sorties semblables, la détection n'est pas toujours immédiate, il faut choisir dans certains cas des séquences adaptées (Cf III.4.1).

Pour étudier la propagation, on peut distinguer trois cas de cheminement dans un graphe lorsque un arc est parcouru :

* détection

C'est le cas le plus favorable, parce que l'état atteint après l'application d'un vecteur d'entrée n'est pas le bon et ne délivre pas la même sortie que l'état normalement attendu.

* non détection

Le défaut n'est pas détecté, l'état atteint même s'il n'est pas celui attendu délivre une sortie identique à celle de ce dernier. Il faut attendre la suite de la séquence pour conclure sur la détection ou la non détection.

* absorption

C'est là le cas le plus défavorable, puisque l'état atteint après l'application d'un vecteur correspond avec l'état attendu. En fait, si la séquence de test s'arrêtait sur ce dernier vecteur, la défaillance ne serait pas détectée.

L'application d'une séquence de test consiste en une succession de vecteurs d'entrée donc d'arcs parcourus qui à chaque fois produisent un de ces trois cas de figure. La probabilité de propagation est une résultante de la probabilité de ces trois cas élémentaires.

III.5.5.1) PROBABILITES ELEMENTAIRES DE PROPAGATION

Chacun de ces trois cas possède une probabilité élémentaire d'apparaitre.

* probabilité d'absorption P_{ab}

A chaque arc parcouru, il existe une chance sur le nombre d'états de la machine (S) pour que l'état atteint soit celui qu'aurait dû avoir celle ci à cet instant, d'où :

$$P_{ab} = \frac{1}{S}$$

* probabilité de non détection P_{nd}

L'état atteint n'est pas le bon mais il délivre une sortie identique à celui normalement atteint.

Soit $\mathcal{S}$: l'ensemble des états possibles

$\mathcal{S}_1$: une partition de $\mathcal{S}$ telle que $\mathcal{S}_1$ constitue l'ensemble des états possibles de la machine correcte.

$\mathcal{S}_1$ peut également être partitionné par rapport aux sorties délivrées par chacun de ses états

$$\mathcal{S}_1 = \Delta_1 \cup \Delta_2 \cup \ldots \cup \Delta_i \cup \ldots \cup \Delta_k$$

k étant le nombre de sorties différentes délivrées par la machine correcte

La probabilité d'atteindre un état $S_j \in \Delta_i$ tel que la sortie soit égale à O_j vaut : $\frac{Card(\Delta_i)}{S}$

La valeur maximale de cette probabilité correspond à la valeur maximale de Card (Δ_i) et correspond à la classe d'états ayant le plus grand nombre d'états à sorties identiques.

D'où la valeur maximale de la probabilité de non détection en excluant l'état absorbant qui fait partie des états ayant une sortie attendue :

$$P_{nd} = \frac{Max\ Card(\Delta_i)}{S} - \frac{1}{S}$$

* probabilité de détection P_d

Il y a détection lorsque l'état atteint ne délivre pas la sortie attendue.

Nous avons $P_d + P_{nd} + P_{ab} = 1$

D'où :

$$\boxed{P_d = 1 - P_{nd} - P_{ab}}$$

Cette valeur est la probabilité minimale de détection pour un vecteur de test appliqué.

Connaissant ces probabilités élémentaires, il est possible de calculer la probabilité de propagation d'une séquence de vecteurs d'entrée.

Nous allons calculer celle ci pour deux cas de séquence de test

- une séquence de longueur K non répétitive avec K>>A
- une séquence de longueur L périodique

III.5.5.2) SEQUENCE NON REPETITIVE

C'est une séquence ayant la particularité de ne faire évoluer la machine que dans une certaine zone et, en fait, m arcs ne sont pas parcourus dans la machine correcte.

La séquence a aussi pour propriété d'éliminer les phénomènes d'absorption dans la mesure où si, à un instant donné, l'état atteint est absorbant et donc le reste du cheminement redevient celui de la machine correcte, à un autre instant le cheminement sera autre et la défaillance sera détectée.

Ceci est possible si la séquence est suffisamment longue (donc chaque état est atteint plusieurs fois) et si elle n'est pas répétitive.

On a alors $P_{ab} = 0$ et $P_{nd} = \frac{\text{Max Card}(\Delta_i)}{S}$

La figure III.14 illustre cette propriété.

Soit une portion de graphe et les éléments suivants:

S_3 et S_5 ont une sortie différente

S_0 est un état de non détection mais n'est pas absorbant

S_4 est un état absorbant

l'arc "d" est dévié

A un instant, la séquence est a_b_c, S_4 étant absorbant, le défaut n'est pas vu.

Plus loin dans la séquence, on aura a_d_e est le défaut sera détecté puisque S_3 et S_5 ont des sorties différentes.

On peut donc dire que ce type de séquence n'absorbe jamais les défauts.

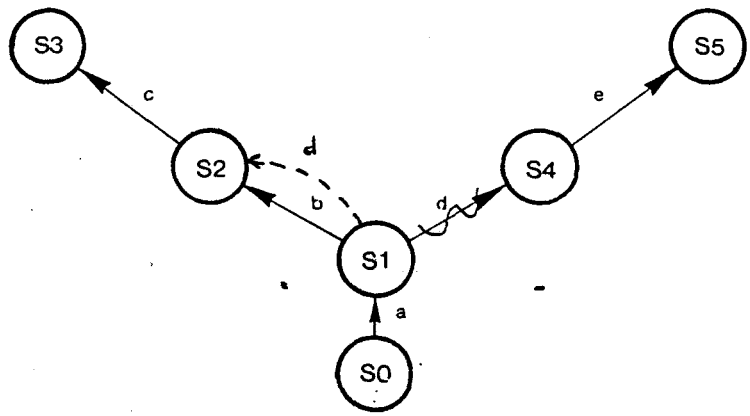


Fig.III.14

On note P_S , la probabilité de succès de la propagation et on peut calculer celle ci en fonction du nombre de vecteurs d'entrée déjà appliqués depuis l'origine de la séquence.

P_{ak} = probabilité de succés du kième vecteur de la séquence

P_{SK} = Probabilité de succés au bout de K vecteurs appliqués

A l'origine $P_A = P_S$

Si nous calculons P_{s2}, la probabilité de succès au bout du 2ième vecteur appliqué :

$$P_{s2} = P_{d1} \cdot P_{dn1} + P_{dn2} \cdot P_{d2} + P_{d1} \cdot P_{d2}$$

$$= P_{d1}(1-P_{d1}) + P_{d2}(1-P_{d2}) + P_{d1} \cdot P_{d2}$$

et comme $P_{d1}=P_{d2}=P_d$ nous avons

$$P_{s2} = P_d (1+P_{nd})$$

En généralisant à une séquence de K vecteurs

$$P_{sK} = P_d (1+P_{nd}+P_{nd}^2+\ldots+P_{nd}^i+\ldots+P_{nd}^{K-1})$$

Soit en fait

$$P_{sK} = 1-(P_{nd})^K = 1-(1-P_d)^K$$

et Q_{sK} qui est le complément à 1 de P_{sK} et qui vaut $Q_{sK}=(P_{nd})^K$ peut d'ailleurs s'obtenir directement en raisonnant avec les probabilités élémentaires de non détection.

Lorsque K tend vers l'infini P_{sK} tend vers 1 puisque $P_{nd} < 1$

Une séquence aléatoire de longueur infinie détecte toute défaillance d'un circuit séquentiel.

Le tableau suivant donne la probabilité d'échec à divers stades de la séquence pour deux machines possédant le même nombre d'états mais ayant une répartition des sorties différentes : 3 vecteurs de sortie possibles pour la première, 2 pour la seconde.

Machine 1: S=8, Card(Δ_1)= 3, Card(Δ_2)= 2, Card(Δ_3)= 3

Machine 2: S=8, Card(Δ_1)= 4, Card(Δ_2)= 4

K	1	2	3	5	10	20	
Q_{sK}	0,37	0,14	$5,2.10^{-2}$	$7,4.10^{-3}$	$5,5.10^{-5}$	$3,0.10^{-9}$	Machine 1
Q_{sK}	0,5	0,25	0,12	$3,1.10^{-2}$	$9,7.10^{-4}$	$9,5.10^{-7}$	Machine 2

Plus le nombre d'états possédant des sorties identiques est important et plus la séquence de détection doit être longue.

III.5.5.3) SEQUENCE REPETITIVE.

Nous considérons ici une séquence de faible longueur L et ayant la particularité d'être répétée .

Nous prendrons les hypothèses suivantes pour effectuer le calcul de la probabilité de propagation.

- on considère le test terminé lorsque la première période a été déroulée et même si par ailleurs une seconde est enchainée
- on prend en compte les phénomènes d'absorption

Les conséquences de ces hypothèses ont pour effets de rendre le calcul plus pessimiste qu'il ne l'est en réalité :

La séquence étant périodiquement répétée, s'il n'y a pas eu détection au bout des L vecteurs appliqués, on peut espérer que la ou les suivantes mettront en évidence la défaillance sauf, si il y a absorption au bout des L premiers vecteurs auquel cas les séquences suivantes de L vecteurs boucleront également: il n'y a jamais détection.

La défaillance est alors latente dans la mesure où le circuit séquentiel délivre des sorties conformes à ce qui est attendu mais en évoluant sur une mauvaise boucle.

Le calcul s'effectue de la même manière que précédemment en introduisant en plus la probabilité d'absorption P_{ab} .

Au bout d'une séquence de 2 vecteurs de la séquence de L la probabilité d'échec vaut:

$$P_{E2} = P_{nd_1} \cdot P_{a_2} + P_{a_1} \cdot P_{a_2} = P_a \, (P_a + P_{nd})^1$$

Pour une séquence de 3 vecteurs:

$$P_{E3} = P_{nd_1} \cdot P_{nd_2} \cdot P_{a_3} + P_{nd_1} \cdot P_{a_2} \cdot P_{a_3} + P_{a_1} \cdot P_{nd_2} \cdot P_{a_3} + P_{a_1} \cdot P_{a_2} \cdot P_{a_3} = P_a \, (P_a + P_{nd})^2$$

et comme $P_d + P_{nd} + P_{ab} = 1$

Pour une séquence de longueur L la probabilité d'échec vaut :

$$P_{EL} = P_a \, (P_a + P_{nd})^{L-1} = P_a \, (1 - P_d)^{L-1}$$

Remarque : La quantité $(1-P_{EL})$ ne représente pas ici la probabilité de succès au bout de L vecteurs appliqués mais la probabilité pour que la machine défaillante n'ait pas un fonctionnement bouclé sur L vecteurs.

On peut la considérer tout de même, et par défaut, comme la probabilité de succès puisque si au bout des L premiers vecteurs, il n'y a pas eu bouclage, les autres périodes de L vecteurs auront de plus en plus de "chance" de propager la défaillance en générant une sortie non attendue d'où :

$P_{SL} = 1-P_a\ (1-P_d)^{L-1}$, P_{SL} sera une limite inférieure de la probabilité de propagation d'une séquence de période L vecteurs.

Le tableau suivant donne les résultats pour les deux machines du paragraphe précédent.

Pour la machine 1: $P_a = \frac{1}{8}$ et $P_{nd} = \frac{3}{8} - \frac{1}{8} = \frac{1}{4}$

Pour la machine 2: $P_a = \frac{1}{8}$ et $P_{nd} = \frac{1}{2} - \frac{1}{8} = \frac{3}{8}$

L	1	2	3	5	10	20	
P_{EL}	0,12	$4,7.10^{-2}$	$1,8.10^{-2}$	$2,5.10^{-3}$	$1,8.10^{-5}$	$1,0.10^{-9}$	Machine 1
P_{EL}	0,12	$6,2.10^{-2}$	$3,1.10^{-2}$	$7,8.10^{-3}$	$2,4.10^{-4}$	$2,4.10^{-7}$	Machine 2

III.5.6) PROBABILITE DE DETECTION.

C'est la probabilité effective qu'une défaillance ayant affecté la machine soit détectée.

Il faut que deux conditions soient réunies :

* la défaillance est localisée dans la partie testée de la machine - probabilité de localisation
* la séquence de test décèle la défaillance en produisant une sortie non attendue - probabilité de propagation.

Ces deux conditions étant simultanément nécessaires, la probabilité de détection Pdétection vaut le produit des deux précédentes et :

$$Pdétection = \left(1 - \frac{m!\,(A-a)!}{A!\,(m-a)!}\right) \cdot \begin{cases} 1-P_a\,(1-P_d)^{L-1} & \text{séquence cyclique de L} \\ \quad \text{ou} & \\ 1-(1-P_d)^{K} & \text{séquence de longueur } K>>A \end{cases}$$

avec $A = S.n$ S nombre d'états de la machine correcte
n nombre d'entrées

III.6) CONCLUSIONS - TAUX D'INSECURITE RESIDUEL.

Le test en ligne des circuits logiques composant un contrôleur est une nécessité pour éviter les pannes latentes et obtenir la sécurité de fonctionnement.

Le test exhaustif de ceux ci conduit, en particulier pour les circuits séquentiels, à la définition de séquences de test excessivement longues et certainement incompatibles avec la fonction remplie par le circuit dans le cadre du contrôleur; ces séquences de test doivent être mixées avec la tâche d'application.

Pour un test exhaustif, le taux d'insécurité résiduel est égal à 0 et, le taux de couverture de pannes garanti par ce test est égal à 1; ceci constitue le cas idéal et répond au concept de sécurité intrinsèque.

Nous nous sommes intéressés au test partiel, soupçonnant la difficulté d'être complet.

Nous avons donc défini une limite inférieure de la probabilité de détection pour une séquence quelconque et une défaillance très pénalisante pour le test, ceci pour les circuits séquentiels.

Cette probabilité, que nous pouvons assimiler au taux de couverture de pannes du test est inférieure à 1 et le taux résiduel d'insécurité λ_{Si} devant en résulter est égal à :

$$\lambda_{Si} = \lambda_s(1-\mathcal{C}) \text{ avec } 0<\mathcal{C}<1$$

Pour la borne inférieure de $\mathcal{C}$, nous avons une sécurité fiabiliste, pour la borne supérieure, une sécurité absolue.

λ_s représente le taux de défaillance horaire du circuit sous test.

La tâche du concepteur consiste à faire tendre le taux de couverture vers la valeur 1 compte tenu :

* des circuits utilisés
  - étude des schémas internes
  - étude des défaillances les plus pénalisantes

* des possibilités d'inclure des séquences de test performantes et compatibles avec la fonction du circuit dans le contrôleur

Ce chapitre constitue une base pour l'évaluation du taux résiduel d'insécurité d'un contrôleur conçu avec des circuits intégrés logiques. Considérant le taux résiduel global λ_{St} comme la somme des taux de chaque circuit

$$\lambda_{St} = \sum_{i=1}^{i=N} \lambda_{Si}(1-\mathcal{C}_i)$$

avec N le nombre de circuits composant le contrôleur.

On note avec regret que ce taux résiduel a pour effet de relier la sécurité au temps par l'intermédiaire du taux de défaillance horaire des circuits et que le respect d'un objectif de sécurité est alors conditionné au remplacement systématique et préventif de chaque constituant du contrôleur, la sécurité se dégradant avec le temps.

En sécurité intrinsèque ceci n'avait pas lieu d'être , sauf pour des raisons de disponibilité puisque la sécurité ne dépendait pas du taux de défaillance des composants.

INTRODUCTION

Dans ce chapitre, nous allons examiner deux moyens de mise en oeuvre de fonctions logiques pour obtenir une certaine sécurité de fonctionnement.

La "logique alternative", déjà évoquée, est une méthode systématique qui crée des fonctions ayant la propriété d'avoir des défauts orientés (informations statiques). L'association de ces fonctions suivant certaines règles permet la propagation des défauts jusqu'aux sorties sans risque d'absorption.

Une seconde méthode, la logique dynamisée, utilise les circuits logiques tels quels; les signaux véhiculés possèdent des relations de phase précises. La sécurité est basée sur l'altération de ces relations par les défaillances des circuits logiques ou par les signaux d'entrée du contrôleur.

Pour cette dernière méthode, nous présentons une réalisation et une évaluation, au moyen des résultats acquis au chapitre précédent, du taux d'insécurité résiduel.

D'autres tentatives ont été effectuées pour obtenir des fonctionnements sûrs avec des circuits logiques, en particulier en exploitant les propriétés d'inversibilité de certaines fonctions (REF.24)(REF.25), les résultats amènent également à la définition d'un indice de sécurité probabiliste.

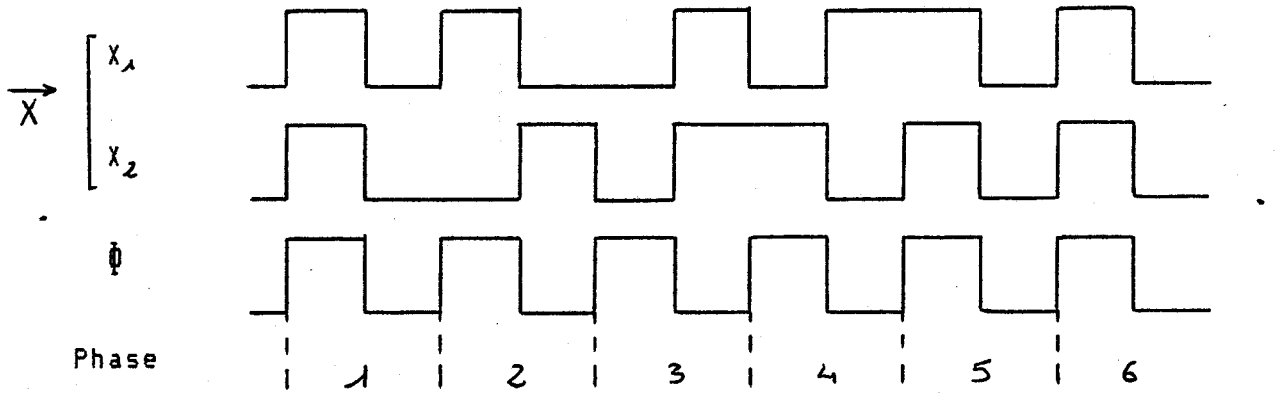
IV.1) LA LOGIQUE ALTERNATIVE.

La logique alternative est une des méthodes permettant l'autotest de contrôleurs conçus avec des circuits logiques.

Son principe consiste à introduire la notion de temps sur chaque entrée de fonction, cette même notion de temps étant retrouvée à la sortie.

La variable alternative X est définie par un couple $(x,\bar{x})$, l'information contenue dans X est donnée par x, $\bar{x}$ représentant la composante test corrélée à x.

La discrimination entre x et $\bar{x}$ est obtenue en comparant X à une horloge Φ synchrone (figure IV.1).



durant les diverses phases, l'information contenue dans $\vec{X}$ vaut

Phase 1 $x_1 = 1$ $x_2 = 1$
Phase 2 $x_1 = 1$ $x_2 = 0$
Phase 3 $x_1 = 0$ $x_2 = 0$
Phase 4 $x_1 = 0$ $x_2 = 1$
Phase 5 $x_1 = 1$ $x_2 = 1$
Phase 6 $x_1 = 1$ $x_2 = 1$

cette information x est repérée par rapport à l'alternance 1 du signal d'horloge Φ .

Fig.IV.1

IV.1.1) FONCTIONS DUALES (REF.22).

La logique alternative ne manipule que des informations alternatives. Pour réaliser une fonction complexe par assemblage de plusieurs fonctions simples, il faut que chacune de celles ci délivre des sorties elles mêmes alternatives et toujours synchrones de Φ .

Les fonctions remplissant de telles conditions sont dites duales:

Soit $F(X)$ telle que $F(\overline{X}) = \overline{F}(X)$.

Une application F arbitraire n'est pas toujours naturellement duale et, il est alors nécessaire de la transformer pour qu'elle acquière cette propriété fondamentale en logique alternative.

Soit en exemple la fonction $F(X)$ avec X un vecteur d'entrée à trois composantes a,b,c.

$F(a,b,c)$ donne un vecteur de sortie à deux composantes s_1, s_2.

a	b	c	s_1	s_2	s_1	s_2	s_1	s_2
0	0	0	0	1	0	1	1	0
0	0	1	0	1	1	0	1	0
0	1	0	0	1	0	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	1	1	0
1	1	0	1	0	0	1	0	1
1	1	1	0	1	0	1	1	0
			$F(X)$		$F(\overline{X})$		$\overline{F}(X)$	

Cette fonction n'est pas naturellement duale $F(\overline{X}) \neq \overline{F}(X)$.

En ajoutant une entrée supplémentaire, l'horloge Φ , telle que:

$$F(X) = F^*(X,\Phi) \qquad (1)$$

et $$\overline{F}(X) = F^*(\overline{X},\overline{\Phi}) \qquad (2)$$

il vient de (1) que $\overline{F}(X) = \overline{F}^*(X,\Phi)$

et donc de (2) que $F^*(\overline{X},\overline{\Phi}) = \overline{F}^*(X,\Phi)$

ceci est la définition même d'une fonction duale.

En reprenant la fonction F(a,b,c) transformée en $F^*(a,b,c,\Phi)$

Φ	a	b	c	s_1	s_2	s_1	s_2	s_1	s_2
1	0	0	0	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	0
1	0	1	0	0	1	1	0	1	0
1	0	1	1	1	0	0	1	0	1
1	1	0	0	1	0	0	1	0	1
1	1	0	1	0	1	1	0	1	0
1	1	1	0	1	0	0	1	0	1
1	1	1	1	0	1	1	0	1	0
0	0	0	0	1	0	0	1	0	1
0	0	0	1	0	1	1	0	1	0
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0	1	0
0	1	0	0	0	1	1	0	1	0
0	1	0	1	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	0	1	0	1
				$F^*(X)$		$F^*(\overline{X})$		$\overline{F}^*(X)$	

Ainsi, si X est une entrée alternative, la sortie S produite par la fonction F^* est également alternative.

$F(X,\Phi)$ s'exprime en fonction de F

$$F^*(X,\Phi) = \Phi\, F(X) + \overline{\Phi}\ \overline{F}(\overline{X})$$

IV.1.2) REPRESENTATION SCHEMATIQUE ET PRINCIPE DE DETECTION.

La fonction duale peut être retranscrite schématiquement à partir de la forme de son équation logique (Fig.IV.2).

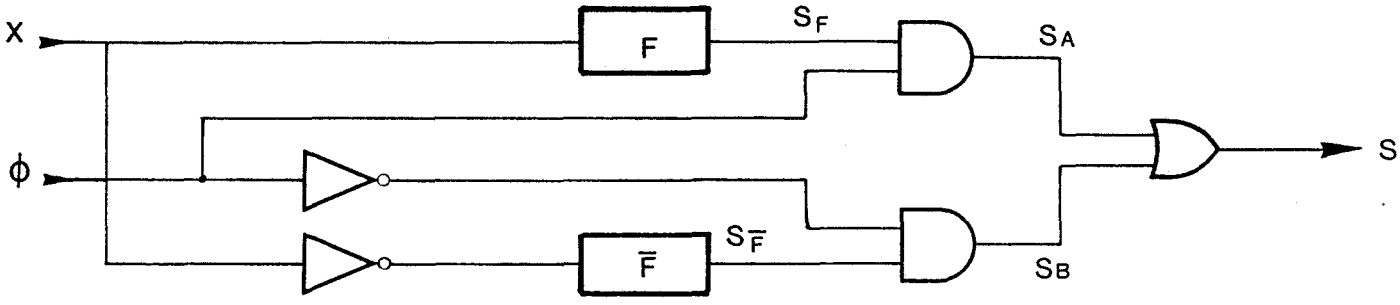


Figure.IV.2

Le principe de la détection est le suivant :

La sortie générée par le circuit est alternative et on peut considérer ce qui peut se produire sur une période du signal d'horloge.

Cette sortie peut présenter quatre formes.

$$S_1 = (s,\bar{s}) \qquad S_2 = (\bar{s},s) \qquad S_3 = (s,s) \qquad S_4 = (\bar{s},\bar{s})$$

Seules S_1 et S_2 constituent des variables alternatives, S_3 et S_4 sont des sorties statiques sur une période de ϕ et révèlent une défaillance dans la branche F ou une défaillance dans la branche $\bar{F}$.

On peut également interpréter ceci en introduisant le codage :

$\mathcal{C}$ est le code de sortie partitionné en $\mathcal{C}_1$ et $\mathcal{C}_2$ avec

$\mathcal{C}_1 = (S_1, S_2)$ et $\mathcal{C}_2 = (S_3, S_4)$

Une sortie appartenant à $\mathcal{C}_2$ est statique sur une période de ϕ.
Une sortie appartenant à $\mathcal{C}_1$ est alternative, donc correcte.

Lorsqu'une défaillance se produit, elle se situe soit dans la branche F, soit dans $\overline{F}$, aussi, si elle est détectée, l'une des branches fournit la même valeur que l'autre, alors que normalement les résultats sont complémentaires d'où une sortie statique sur une période d'horloge.

Il se peut que certains vecteurs d'entrée ne révèlent pas la défaillance, la valeur fournie par la branche en panne est alors correcte, la variable de sortie est alternative et la panne latente.

Selon le type de panne apparue et le vecteur d'entrée appliqué, la sortie est hors code attendu, $\mathcal{C}_2$, ou correcte $\mathcal{C}_1$.

Dans la théorie des circuits manipulant des informations codées, on introduit des propriétés particulières que doivent satisfaire les circuits pour être sûrs. La logique alternative peut être analysée avec cette démarche.

Sûreté de fonctionnement en présence d'une panne (Fault secure)

Un circuit est sûr en présence d'une défaillance $\mathcal{L}$ lorsque la sortie produite est : soit la sortie attendue $(\in \mathcal{C}_1)$
soit une sortie hors code $(\in \mathcal{C}_2)$

Autotestabilité

Un circuit est autotestable pour une défaillance $\mathcal{L}$ lorsqu'il existe un vecteur d'entrée $\vec{X}_d$ qui produit une sortie hors code $(\in \mathcal{C}_2)$.

Un circuit possédant ces deux propriétés est dit totalement autotestable.

Ainsi, s'il n'existe pas de vecteur X_d pour une panne $\mathcal{L}$, celle ci est non détectée et il se peut alors qu'une seconde panne $\mathcal{B}$, apparaissant dans le circuit, produise une sortie fausse.

Une sortie est fausse lorsqu'elle appartient au code de sortie attendu mais qu'elle est le contraire de ce qu'elle aurait dû être.

IV.1.3) MECANISME D'APPARITION D'UNE SORTIE FAUSSE.

Explicitement, une sortie fausse se présente ainsi :

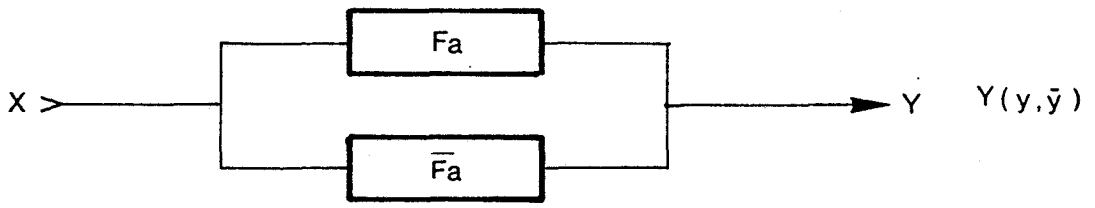
X = (0,1) au lieu de (1,0) ou encore
X = (1,0) au lieu de (0,1)

Comme on peut le voir, la sortie fausse possède toutes les caractéristiques d'une variable alternative et ne peut donner lieu à une détection.

L'existence de telles sorties est liée à la nature des fonctions alternatives : deux branches F_i et $\bar{F}_i$ produisant respectivement x_i et $\bar{x}_i$.

On montre facilement que ces sorties fausses sont produites en présence de deux défaillances.

Soient α et β deux défaillances et la représentation synthétique d'une fonction alternative :



Apparition de α $\quad F_a \longrightarrow F_{a\alpha}$

Il existe des vecteurs $\vec{X}_d$ tels que $Y = (\bar{y}, \bar{y})$ qui détectent α

Il existe également des $\vec{X}_{nd}$ tels que $Y = (y, \bar{y})$ qui ne détectent pas α.

Apparition de β dans la branche $\bar{F}_a$

Il peut exister un vecteur X_f tel que celui-ci détecte la défaillance α de F_a et détecte la défaillance β dans la branche $\bar{F}_a$. Les deux branches donnent une valeur incorrecte et la sortie alternative produite est fausse.

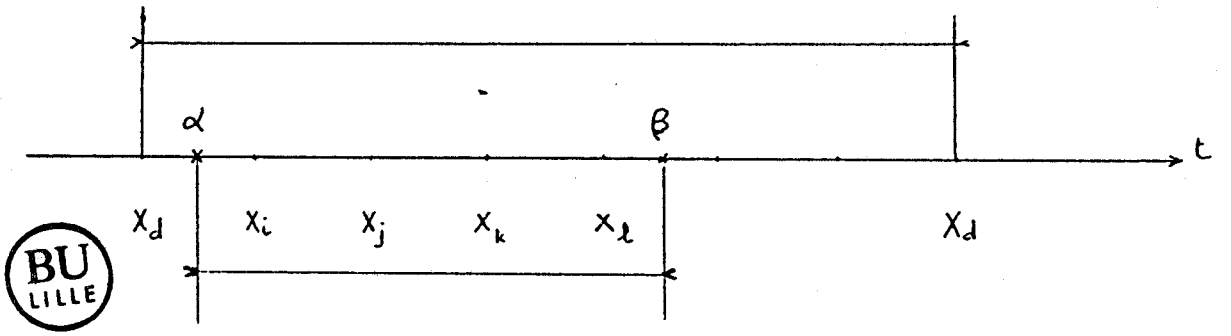
On retrouve la même chose avec une cascade de fonctions alternatives car la seconde défaillance peut ne pas être localisée sur le circuit qui est affecté par la première et provoquer tout de même une sortie fausse.

L'occurence de sorties fausses est due aux faits suivants :

- il n'existe pas de vecteurs $\vec{X_d}$ capables de détecter la première défaillance

- les $\vec{X_d}$, quand ils existent, ne sont pas appliqués suffisamment tôt (présence occasionnelle du processus dans certaines zones de son domaine d'évolution contenant les $\vec{X_d}$).

le temps séparant deux apparitions de ceux ci est supérieur au temps séparant l'apparition de la défaillance β après α.



si X_i, X_j, X_k, X_l ne révèlent pas α, à partir de l'instant d'apparition de β il y a un risque d'obtenir une sortie fausse.

IV.1.4) CONCEPTION DES ASSEMBLAGES EN LOGIQUE ALTERNATIVE.

Chaque porte, plus ou moins complexe, est désormais conçue pour produire un défaut orienté (sortie continue) en présence d'une panne.

Chaque porte logique est également conçue pour réagir à une entrée continue en produisant une sortie continue.

Lorsqu'on veut assembler ces portes pour réaliser une fonction plus complexe, il faut prendre certaines précautions d'assemblage afin d'éviter les problèmes d'absorption des défauts générés par ailleurs (Cf. II.5.3). En logique alternative ces précautions sont définies.

En logique alternative, il faut essentiellement s'assurer que le nombre d'inverseurs (ou de portes assurant une inversion) conduisant de la sortie d'une porte jusqu'à la sortie du circuit soit de même parité quel que soit le chemin parcouru depuis celle ci (REF.14).

Cette précaution couvre le dernier cas évoqué au paragraphe II.5.3 lorsqu'il y a risque d'absorption d'un défaut par une reconvergence de plusieurs signaux issus d'une même origine et ayant subis des traitements par des branches différentes.

De cette manière aucune information hors code ne risque de se "perdre" dans la circuiterie.

Ceci étant posé, il reste à vérifier les risques liés à la génération de sorties fausses.

IV.1.5) INSECURITE RESULTANT D'UNE PANNE LATENTE.

Une défaillance et un test insuffisant sont les causes d'une panne latente dans le contrôleur.

IV.1.5.1) TEST INSUFFISANT.

Le test est insuffisant parce que :

* certains vecteurs d'entrée ne sont pas appliqués au contrôleur ou à certaines portes élémentaires le constituant.

* certains vecteurs d'entrée ont une période d'apparition trop longue et il est difficile de pouvoir les considérer comme susceptibles de participer au test . Le test est alors partiel.

Les possibilités d'avoir une panne latente, donc d'avoir une situation dangereuse par la production d'une sortie "fausse", ne sont alors plus à exclure et il faut en évaluer la probabilité d'occurrence.

IV.1.5.2) DEFAILLANCES D'UNE PORTE EN LOGIQUE ALTERNATIVE.

La figure IV.3 représente la structure d'une porte et repère chacun de ces constituants.

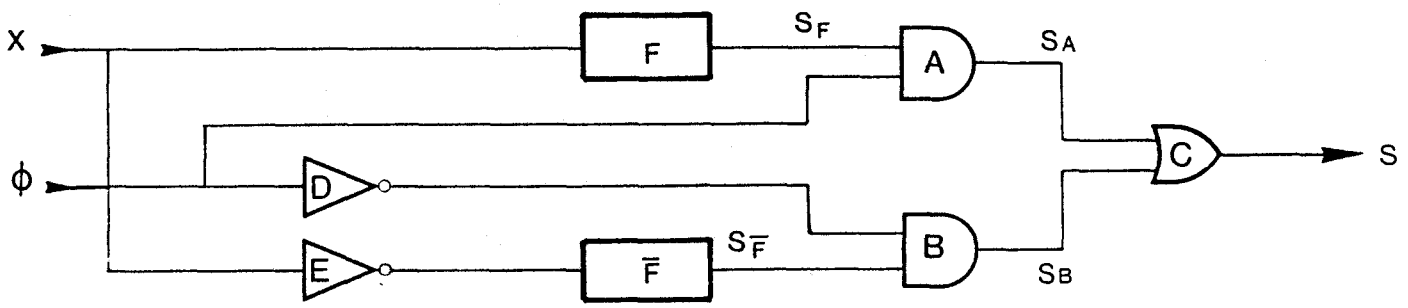


Fig.IV.3.

L'examen de chaque élément va permettre de situer les parties testées et mal testées.

* porte OU de sortie :(C)

Cette porte ne reçoit jamais l'entrée 1.1 puisque les signaux ϕ et $\bar{\phi}$ sur les portes ET précédentes rendent ceci impossible.

Une cause de génération de sortie fausse serait :
une défaillance de C fait délivrer un 0 sur sa sortie lorsque, à la suite d'une autre défaillance, la combinaison 1.1 lui est appliquée.

Cette première défaillance est impossible sans que les trois autres combinaisons (normales) ne révèlent la panne dès son apparition.
La porte OU ne peut donc être à l'origine d'une panne latente.

* portes ET : (A et B)

Ces portes ont un fonctionnement qui dépend des valeurs délivrées par les fonctions F et $\overline{F}$, elles mêmes sollicitées par les valeurs appliquées à l'entrée de la porte alternative.

Durant l'alternance Φ=1 la sortie est délivrée par S_A , durant Φ=0 la sortie est délivrée par S_B. Les défaillance affectant la porte A pendant Φ=0 et la porte B pendant Φ=1 peuvent être la cause de sorties fausses.

La coupure de liaison de Φ sur la porte A, n'est pas toujours détectée lorsque la sortie est du type S=(0,1), on peut alors imaginer le scénario suivant :

- la sortie normale est S(0,1)

- la défaillance intervient, coupure de Φ, elle n'est pas détectée S(0,1)

- une seconde défaillance apparait : l'ouverture de la liaison Φ sur la porte B

- il est possible de trouver des vecteurs d'entrée qui vont donner S = 0 durant Φ=1 ,S = 1 durant Φ=0 et S =1 durant Φ=1, S = 0 ou 1 durant Φ=0, ceci ne dépend que des vecteurs d'entrée appliqués , la sortie délivrée est alors (1,0) au lieu de (0,1) d'où une sortie fausse.

* inverseurs : (D et E)

Ces inverseurs sont sollicités en permanence sur leurs deux états, une défaillance les affectant est immédiatement détectée ou si ce n'est le cas est détectée au premier changement de la variable de sortie : passage de S = (0,1) à S = (1,0).

* fonctions F et $\overline{F}$:

Ces fonctions pouvant être plus complexes que de simples portes à deux entrées, elles risquent plus d'être insuffisamment testées et c'est sur celles ci que doit porter essentiellement l'attention du concepteur.

Le calcul du taux résiduel d'insécurité résultant d'un test insuffisant peut être effectué avec les résultats obtenus au chapitre précédent sur les fonctions combinatoires (paragraphe III.2).

IV.1.6) CONCLUSION.

Des études rigoureuses (REF.22) de cette méthode existent, notre propos a été essentiellement d'en exposer le principe et d'observer les conditions d'apparition de situations contraires à la sécurité : les défauts affectant simultanément les signaux aux instants Φ et $\overline{\Phi}$ consécutifs.

Ces situations sont liées aux possibilités de pannes latentes dues à un test insuffisant :

* vecteurs d'entrées non générés par le processus

* récurrence du test insuffisante.

On émet alors généralement l'hypothèse suivante :

Le temps séparant l'apparition d'une première panne et l'application du vecteur de test qui doit la détecter est inférieur au temps séparant cette même défaillance et une seconde qui génèrerait une sortie fausse(REF.21).

La démarche consiste à estimer sous forme d'une probabilité ces deux temps et à calculer la probabilité d'occurrence de l'évènement contraire à la sécurité.

Notre démarche, par les calculs de taux de couverture du test est quelque peu différente puisqu'elle consiste à estimer la probabilité de présence de pannes latentes.

D'un point de vue du domaine d'application, la logique alternative semble intéressante pour manipuler des grandeurs d'entrée non prévisibles, elle peut permettre ainsi de réaliser des comparateurs ou des voteurs : la manipulation d'informations bit à bit étant aisée.

L'annexe 5 donne l'exemple d'une architecture de voteur avec cette méthode.

IV.2) LA LOGIQUE DYNAMISEE.

La logique dynamisée est une forme moins structurée que la logique alternative, puisque cette dernière utilise la puissance de détection des fonctions duales pour produire des défauts orientés.

La logique dynamisée ne possède pas une aussi grande rigueur, par contre la très grande liberté que procure son mode de dynamisation, lui permet des tâches d'analyse de signaux prévisibles et cycliques.

A chaque instant, les informations reçues, traitées et générées sont connues.

L'exemple traité dans ce chapitre est celui d'un contrôleur de signal de chien de garde. En particulier , il sert à analyser la conformité du signal équitemps défini au chapitre I.

Afin de préciser les performances attendues d'un tel contrôleur, nous donnons son cahier des charges. On constate alors que le problème n'a de solution satisfaisante et fonctionnellement simple qu'avec la logique dynamisée.

IV.2.1) CAHIER DES CHARGES DU CONTROLEUR.

Le signal équitemps, noté S_{ξ}, est la matérialisation extérieure et synthétique du bon fonctionnement d'un microprocesseur soumis à un test comportemental.

Ces caractéristiques, hors défaillance, sont très simples :

période $T_0 = 200\mu S$

rapport cyclique $r = 0,5$

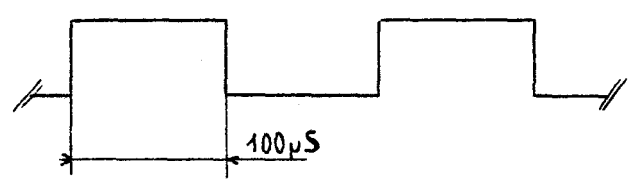


Fig.IV.4

Lorsqu'une défaillance est détectée par le test comportemental, ces caractéristiques peuvent se modifier de plusieurs manières:

- changement de rapport cyclique
- modification de la fréquence
- présence de transitions parasites

Ces anomalies, qui constituent des défauts à détecter, sont permanentes ou fugitives.

On admet toutefois une tolérence sur le rapport cyclique ne pouvant excéder $\pm 0,5\mu S$.

La tâche demandée au contrôleur est la suivante:

- contrôler que chaque demi période fait $100\mu S \pm 0,5\mu S$
- vérifier l'absence de transitions parasites
- mémoriser toute altération même fugitive
- générer en sécurité un signal dynamisé de fréquence $\frac{f_o}{2}$ lorsque le contrôle ne révèle aucun défaut.

Le contenu de ce cahier des charges renseigne sur la nature du travail que doit effectuer le contrôleur. L'information traitée n'est pas une variable booléenne classique, c'est la forme du signal S_2 qui contient l'information intéressante.

La tâche est une analyse de forme du signal équitemps suivie d'une décision binaire :

* forme attendue $\longrightarrow$ dynamisation à $\frac{f_o}{2}$
* altération détectée $\longrightarrow$ arrêt de dynamisation irréversible

En plus de cette aspect fonctionnel, l'aspect sécurité conditionne la réalisation pratique et donc, toute défaillance du contrôleur lui même doit avoir le même effet que la détection d'un défaut sur le signal objet du contrôle $\longrightarrow$ arrêt irréversible (REF.23).

IV.2.2) ASPECTS FONCTIONNELS.

Il n'est pas possible de définir des règles de conception visant à réaliser une fonction donnée, d'une part parce que chaque cahier des charges est un cas d'espèce et d'autre part parce qu'une même fonction admet généralement plus d'une solution réalisable.

Toutefois, certains critères permettent intuitivement de choisir, à performances fonctionnelles égales, parmi plusieurs contrôleurs :

- nombre réduit de composants :

Le contrôleur n'est qu'une circuiterie annexe au processeur, il ne doit pas rivaliser en encombrement ou en complexité dans son analyse.

Une réalisation simple donne généralement une disponibilité meilleure par un nombre de sources de pannes réduit, il n'est pas concevable que le processus soit constamment arrêté par une défaillance de son contrôleur.

- l'aspect sécurité du contrôleur :

Le choix de composants particuliers et peu testables en ligne est à éviter. La stratégie adoptée pour l'analyse du signal doit tenir compte, dans la mesure du possible, des contraintes de sécurité et donc des contraintes de test de la structure retenue.

IV.2.3) ASPECT SECURITE.

La sécurité repose sur le respect de deux points :

- le test des circuits utilisés :

Ceci pour s'assurer que la fonction remplie localement par un des éléments de l'assemblage est la bonne.

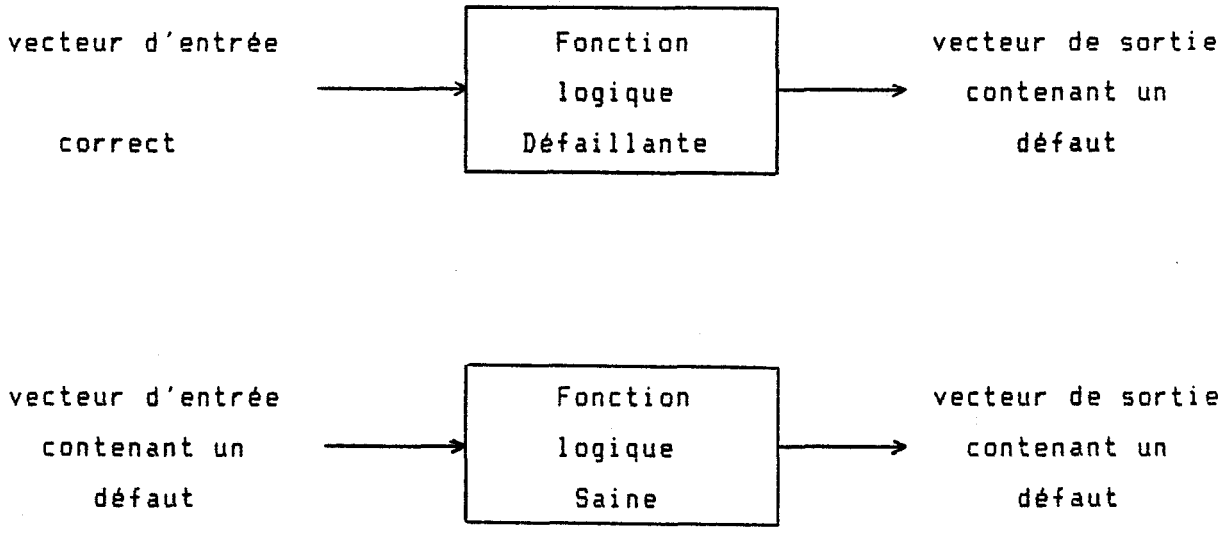
Pour être certain qu'ils n'engendreront pas de fonctionnement permissif en cas de défaillances se situant en amont d'eux (absorption de défaut due à une panne latente)

- propagation des défauts :

Un défaut généré par les circuits en amont de celui considéré doit pouvoir être correctement propagé jusqu'à la sortie effective du contrôleur et provoquer ainsi un arrêt de la dynamisation.

Test et propagation peuvent être représentés localement de la manière suivante :

En isolant une fonction logique de son contexte



IV.2.3.1) TEST DES CIRCUITS LOGIQUES.

Le test des circuits logiques a été étudié au chapitre précédent, tous les éléments sont connus pour estimer la qualité du test obtenu dans le cadre d'un contrôleur donné.

Le fonctionnement est ici du type cyclique, il est donc nécessaire que durant une période, les signaux proposés à chaque circuit séquentiel ou combinatoire soient suffisamment étoffés pour rendre ce test significatif.

Le concepteur est tributaire de la longueur de la période de récurrence des signaux d'entrée pour "installer" les séquences de test.

Ceci exclut naturellement :

- les fonctions combinatoires complexes, celles qui possèdent un grand nombre d'entrées

  (le nombre de vecteurs de test est doublé par entrée ajoutée)

- les fonctions séquentielles à grand nombre d'états donc à grand nombre d'arcs

  (le nombre d'arcs d'une machine séquentielle, avec la représentation adoptée, est égal au produit du nombre d'états par le nombre d'entrées).

IV.2.3.2) PROPAGATION DES DEFAUTS.

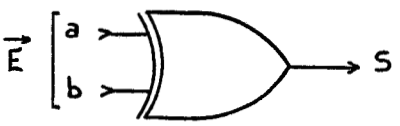
Lorsqu'un défaut, matérialisé sur un vecteur, est présent à un instant, il a pour origine soit une anomalie constatée sur le signal à analyser, soit une défaillance d'un des circuits testés.

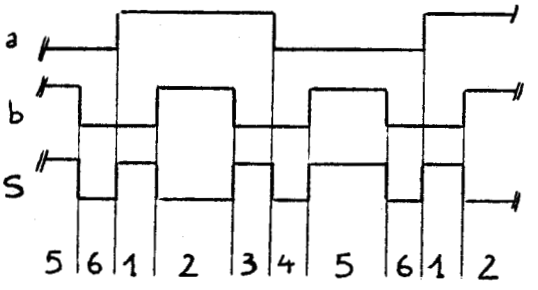
La propagation consiste à conserver, malgré les transformations subies par ce défaut au cours de son passage dans les autres circuits, sa nature d'information "défaut".

Pour étudier le mécanisme de propagation, nous sommes aidés par le caractère cyclique des signaux. Il est ainsi possible de représenter pour chaque circuit et pour son vecteur d'entrée, l'ensemble des phases successives de fonctionnement.

Numéro de phase	Vecteur d'entrée	Vecteur de sortie
1	E1	S1
2	E2	S2
.		
.		
.		
j	E_j	S_j
k	E_j	S_j
.		
.		
.		
n	E_n	S_n

Le passage d'une phase à une autre est lié au changement du vecteur d'entrée. Chaque phase dure un temps déterminé et la somme de ces durées correspond à la période du signal du vecteur d'entrée du circuit. La figure IV.5 donne un exemple concret.





phase	$\vec{E}$ ab	S
1	10 (01)	1 (1)
2	11	0
3	10 (11)	1 (0)
4	00	0
5	01	1
6	00	0

si durant la phase 3, b conserve sa valeur 1, la sortie vaut 0 au lieu de 1, il y a propagation du défaut de b

Fig.IV.5

Cette représentation permet d'avoir un suivi précis de la propagation au niveau d'un circuit logique, il suffit d'introduire les défauts sur les vecteurs d'entrée :

- modification de l'état d'une entrée durant une phase
- modification de l'état d'une entrée durant plusieurs phases
- modification de l'état de plusieurs entrées

Si la valeur du vecteur de sortie est modifiée durant une ou plusieurs phases, le défaut est alors propagé, si tous les circuits propagent le défaut, il est détecté.

La propagation est liée à la fonction remplie par le circuit logique considéré et à l'état du vecteur d'entrée au cours d'une phase.

La fonction est réceptive si le défaut provoque un changement par rapport au vecteur de sortie attendu, dans la négative, le défaut n'est pas vu et le vecteur de sortie est inchangé.

Dans l'exemple de la figure IV.5, la fonction est réceptive au défaut de b dans la phase 3, elle ne l'est pas dans la phase 1.

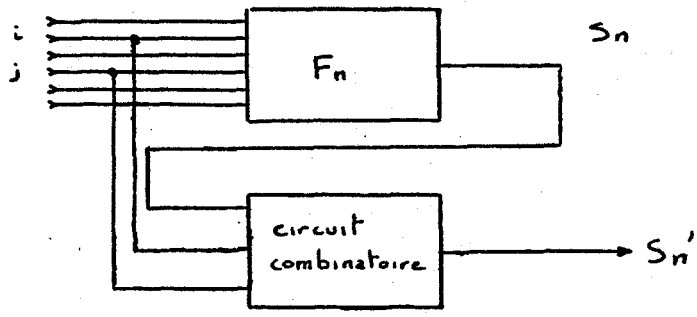
Lorsqu'à un endroit, la propagation est mauvaise (absorption), plusieurs solutions peuvent être envisagées :

- modification générale de la fonction considérée

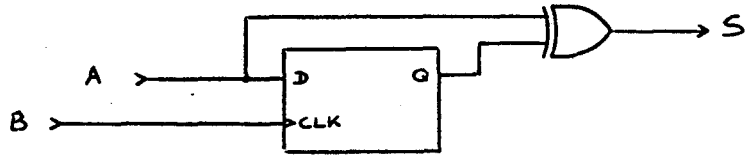
- modification des circuits en amont pour changer la présentation du défaut

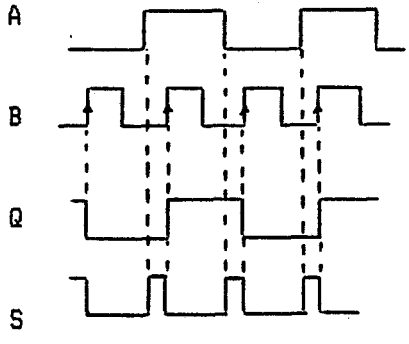
- création de circuits simples combinant les sorties de la fonction concernée et les entrées contenant les défauts non propagés, ce qui peut être assimilé à un pontage de la fonction (Fig.IV.6)

- report des entrées aux défauts non propagés sur d'autres circuits du contrôleur , ce qui peut être vu comme une mise en parallèle de la propagation du défaut (Fig.IV.7)



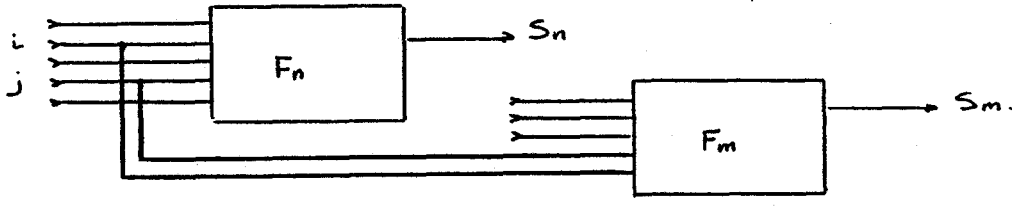
le circuit additionnel est conçu de manière à conserver les informations de l'ancienne sortie tout en incluant les défauts de i,j initialement non propagés.



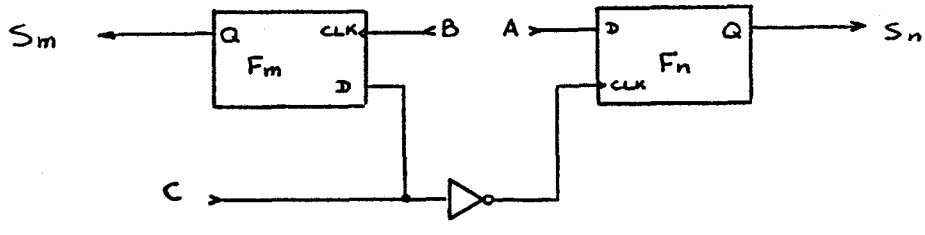


grâce à l'adjonction du OU exclusif, tous les défauts apparaissant sur l'entrée A sont reportés sur la sortie S. L'entrée D de la bascule propageant médiocrement les défauts

Figure IV.6



les défauts initialement propagés par F_n le sont toujours; F_m se charge de propager ceux véhiculés par i et j.



la bascule F_m ne propage pas tous les défauts de l'entrée C (entrée donnée d'une bascule D), par l'intermédiaire d'un inverseur, elle sert d'entrée horloge d'une seconde bascule F_n.

Figure IV.7

IV.2.3.3) TOLERANCE AUX DEFAUTS OU A LEUR PROPAGATION.

Idéalement, un contrôleur tel que celui défini au début de ce paragraphe ne doit admettre qu'un seul mode de fonctionnement permissif (génération de la sortie dynamisée).

Au niveau des signaux, cela signifie que toutes les phases auront une durée determinée et que tout écart traduisant une défaillance conduira à la détection (prise de l'état de sécurité).

Dans la pratique, plusieurs faits amènent à remettre en cause cette définition du contrôleur parfait.

En premier lieu, le cahier des charges admet des tolérances quant à l'analyse du signal (fourchette de $\pm 0,5\mu S$ laissée au microprocesseur).

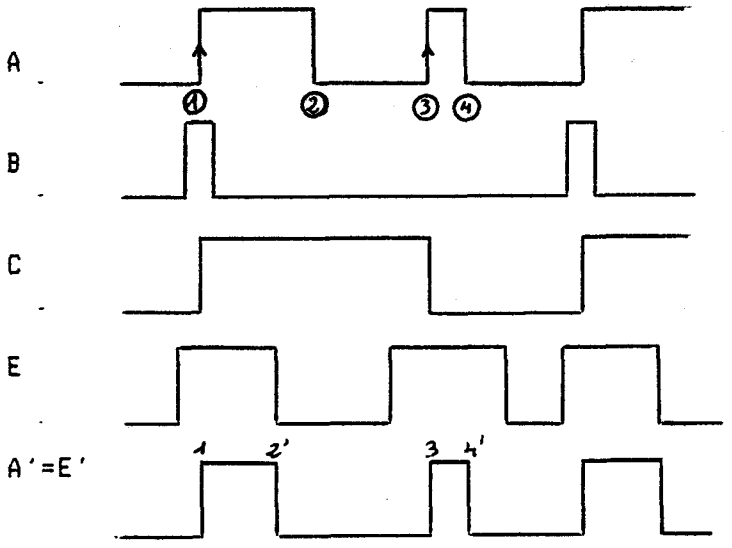
le traitement ainsi que les signaux élaborés peuvent se mouvoir à tout niveau du contrôleur dans une fourchette proportionnelle à la tolérance.

En second lieu, les circuits utilisés ne sont jamais parfaits, (tolérances du constructeur, vieillissement) des phénomènes de dérives vont apparaitre et il serait utopique de vouloir les détecter systématiquement , la gène occasionnée par l'indisponibilité presque permanente serait supérieure au gain de sécurité.

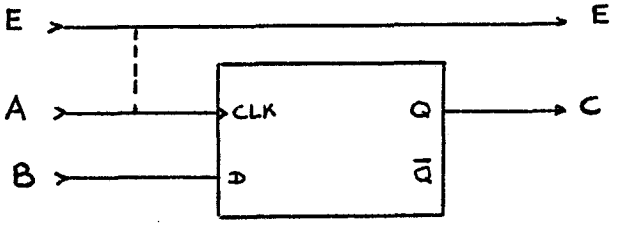
En dernier lieu, des phénomènes tels que les parasites électromagnétiques, inhérents aux environnements industriels, vont altérer les signaux, il serait là également illusoire de vouloir les maitriser exhaustivement.

Pour ces raisons, et moyennant certaines précautions (en identifiant l'origine technologique possible du défaut et en évaluant les conséquences d'une aggravation de celui ci) , on doit admettre l'absorption de certains défauts.

L'exemple suivant montre une défaillance et sa conséquence que l'ont peut tolérer.



les transitions 1 et 3 contiennent l'information importante.



les transitions 2 et 4 sont nécessaires pour conditionner 1 et 3 en transitions montantes

si on considère le signal E et un court circuit entre une ligne véhiculant E et une autre véhiculant A, la résultante est un signal A'=E' dépendant de la technologie utilisée.
en supposant une technologie telle que le signal qui l'emporte est celui qui est à 0 (REF.13), la table de vérité du court circuit est la suivante :

A =0	E =0	A'=E'=0	le court-circuit est équivalent à un ET
A =0	E =1	A'=E'=0	
A =1	E =0	A'=E'=0	
A =1	E =1	A'=E'=1	

la sortie C n'est pas altérée par ce court circuit, la bascule D absorbe le défaut de A (A'). Si l'information E n'a pas de contrainte par ailleurs, le défaut absorbé par cette bascule peut être toléré.

En conclusion, on remarque que la logique dynamisée appliquée à la réalisation de contrôleur de sécurité n'est pas une méthode très structurée.

Le nombre de paramètres à prendre en compte pour assurer la fonctionnalité (cahier des charges), le test des circuits utilisés et la propagation des défauts constatés font qu'énoncer des règles de conception n'est pas possible. Tout au plus, peut on attirer l'attention du concepteur sur les problèmes qu'il a à résoudre et les moyens de vérifier à postériori sa réalisation.

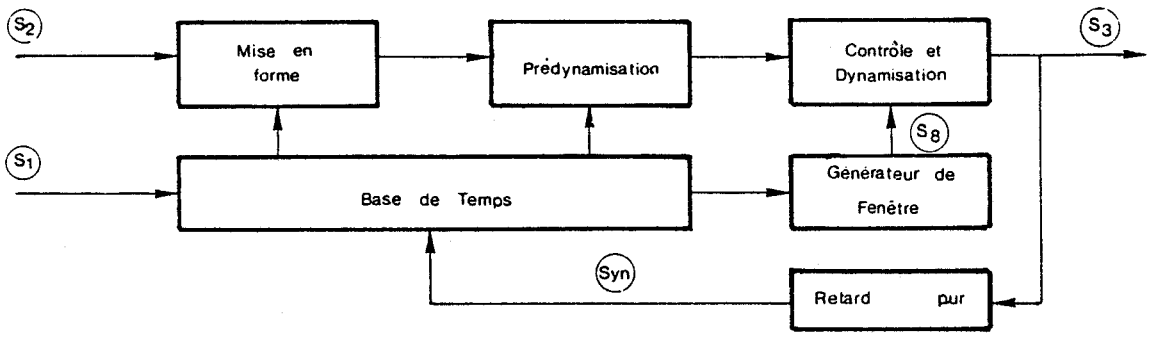
Chaque contrôleur est un cas particulier où les problèmes se posent de façon spécifique, en fonction des circuits utilisés et des signaux que ceux ci vont véhiculer.

Le paragraphe suivant donne un exemple de réalisation pour le contrôleur de signal équitemps.

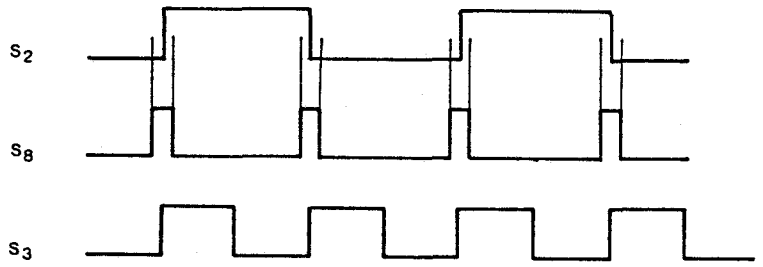
IV.2.4) EXEMPLE DE REALISATION EN LOGIQUE DYNAMISEE (ANNEXE.4)

Le contrôleur décrit ici répond au cahier des charges du paragraphe IV.2.1, il est réalisé à l'aide de circuits logiques de la série 74XX.

Le schéma fonctionnel est le suivant :



Le fonctionnement simplifié est donné par les relations entre les signaux suivants :



L'absence d'un front montant ou descendant de S_2 à l'intérieur d'une fenêtre S_8 provoque l'arrêt de la dynamisation S_3.

IV.2.4.1) SOUS FONCTIONS DU CONTROLEUR.

La base de temps:

C'est une cascade de diviseurs logiques qui fournit aux quatre autres blocs un ensemble de signaux synchrones des informations issues du microprocesseur. Ceci permet d'avoir en permanence une palette de signaux, dits de service, nécessaires à la dynamisation des autres fonctions.

La particularité de cette base de temps est de posséder naturellement un cycle de 128µS et donc de délivrer au générateur de fenêtre des impulsions à cette période. Le contrôle réclamant des impulsions espacées de 100µS, elle nécessite donc , régulièrement, à un instant bien précis d'une réinitialisation effectuée par le signal de sortie.

Ceci constitue, en quelque sorte, un système bouclé et assure la mémorisation du premier défaut constaté.

Le générateur de fenètre :

Ce générateur, constitué d'une bascule D et d'une porte OU exclusif, fournit à partir de la base de temps une fenètre calibrée de 1µS toutes les 100µS. Celle ci doit contenir les transitions de S_2 pour permettre la génération d'un signal dynamisé en sortie.

La mise en forme :

Cet étage est chargé de conserver tels quels les fronts montants du signal équitemps et de transformer les fronts descendants de celui ci en fronts montants. Il ramène ainsi la période du signal S_2 à 100µS alors qu'elle est initialement du double.

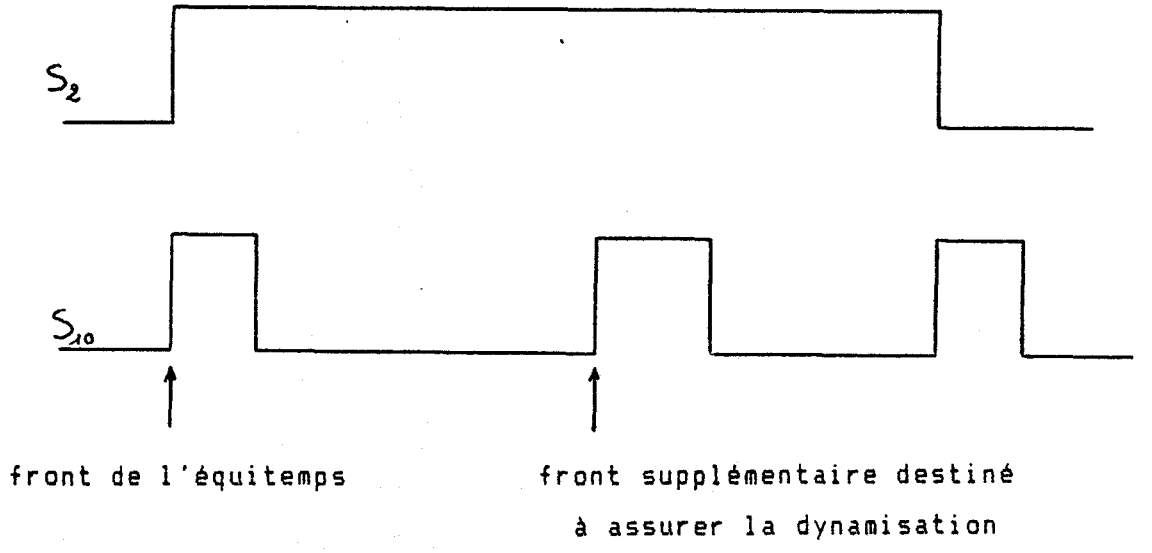
Il permet également d'appréhender par construction toutes les transitions parasites traduisant un défaut sur S_2 .

Ce module est aussi formé d'un OU exclusif et d'une bascule D.

La prédynamisation :

Identique au précédent, ce sous ensemble effectue le même travail en transformant les fronts descendants en fronts montants.

Le signal obtenu est alors le suivant :



Le contrôle et la dynamisation :

Il vérifie la coincidence du front de l'équitemps et de la fenètre calibrée, le signal de sortie passe alors à 1.

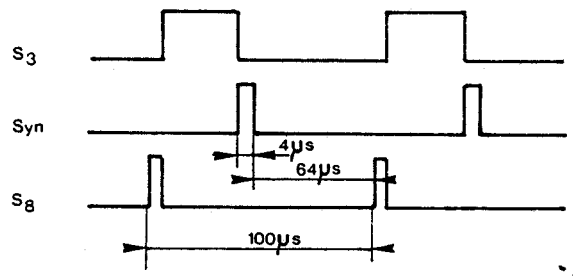
Le front supplémentaire vérifie que la fenètre est refermée et remet la sortie à 0.

Ce front supplémentaire situé à un instant bien précis après celui de l'équitemps va servir à la réinitialisation de la base de temps.

Cette étage est constitué par un motif identique aux quatre précédents.

Le retard pur :

Constitué d'une porte ET et d'une cellule R-C, il provoque à la descente du signal dynamisé de sortie (S_3) une impulsion calibrée de durée égale à 4µS (temps absolu indépendant de la base de temps) qui stoppe la base de temps , la réinitialise et la relance à la fin de ce temps . Celle ci est alors en mesure de fournir 64µS plus tard une impulsion de fenêtre qui encadrera les transitions du signal de l'équitemps en dehors de défaillances.



Tous les défauts constatés sur le signal à contrôler ou sur le contrôleur ont pour effet :

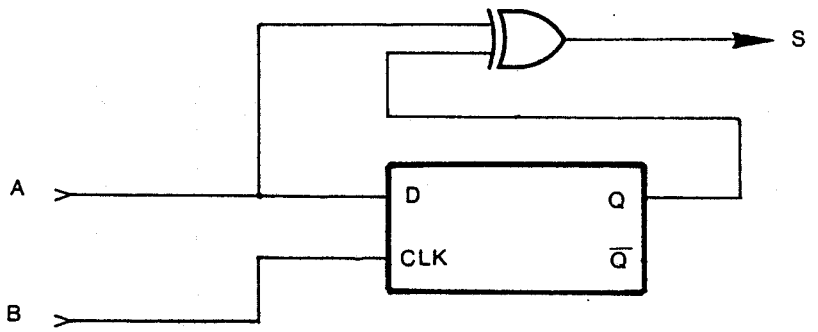
- soit de modifier la position du front descendant du signal de sortie et donc de désynchroniser la fenètre
- soit de faire bouger la fenètre directement
- soit de modifier la position des fronts montants à contrôler

La conséquence est toujours la désynchronisation du contrôleur suivi de l'arrêt de la dynamisation.

IV.2.4.2) CELLULE DE BASE.

La structure de chaque bloc est agencée autour d'une cellule de base, très simple, qui permet de garantir un maximum de propagation des défauts , aussi bien au niveau du vecteur d'entrée que de la cellule elle même.

Son schéma logique est le suivant



La bascule D mémorise sur sa sortie Q , l'état de la variable donnée D à l'instant du front montant de l'entrée horloge CLK.

La sortie Q est recombinée avec l'entrée donnée par un OU exclusif pour fournir une sortie S.

La porte OU exclusif propage intégralement tout défaut sur une entrée, il faut que deux entrées soient altérées pour qu'il y ait absorption.

La bascule D possède une grande activité par son entrée d'horloge CLK , par contre l'entrée donnée n'est active que relativement à CLK (front montant), aussi est elle dérivée vers le OU exclusif.

Cette cellule ne saurait constituer un motif général applicable à tout contrôleur mais, ses qualités et sa simplicité en font un exemple de schéma de cellule type de circuit de sécurité.

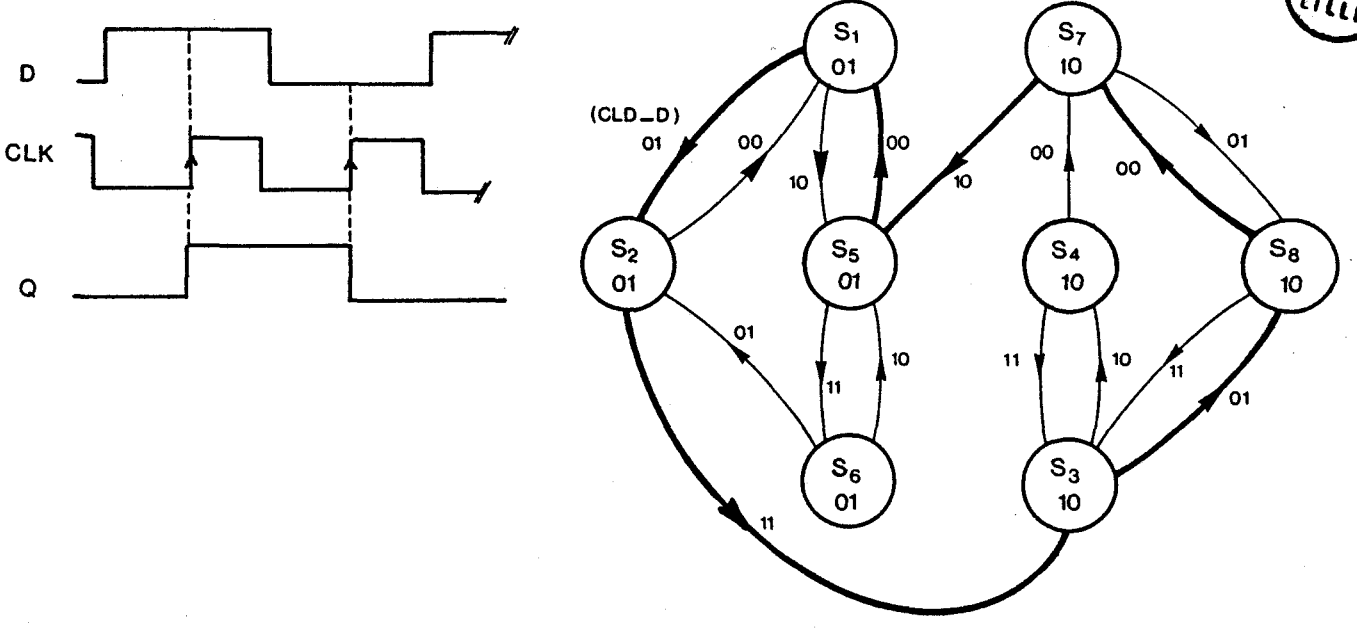
IV.2.4.3) TAUX RESIDUEL D'INSECURITE.

L'étude de sécurité menée à travers de l'analyse du test des circuits utilisés montre que ce sont les quatre bascules D qui doivent faire l'objet d'une quantification de l'efficacité de leur test.

Les portes combinatoires à deux entrées sont correctement testées ainsi que la base de temps où toute défaillance risquant de compromettre la sécurité du contrôleur est détectée.

L'insécurité résiduelle est localisée sur les fonctions séquentielles que sont les bascules D.

Nous allons développer pour l'une d'elle, la bascule de mise en forme, le calcul qui conduit à chiffrer cette insécurité. La figure IV.8 donne les signaux traversant celle ci, sa séquence de test et le graphe équivalent.



m = 10 pour 16 arcs possibles, le test est très imparfait

Fig.IV.8

On peut d'ailleurs calculer la probabilité de localisation $P_{a,10}$ en fonction de "a" qui caractérise la panne la plus difficile à détecter (nombre d'arcs déviés par cette panne = a).

En représentant $Q_{a,10} = 1 - P_{a,10}$

a	2	3	4	5	6	7	10
$Q_{a,10}$	$3,7.10^{-1}$	$2,1.10^{-1}$	$1,1.10^{-1}$	$5,8.10^{-2}$	$2,6.10^{-2}$	$1,0.10^{-2}$	$1,2.10^{-4}$

La probabilité de propagation dépend quant à elle de la longueur de la séquence cyclique L = 6.

Cette valeur de L est trouvée en examinant les signaux d'entrée et en comptant le nombre de transitions sur une période.

On peut calculer les probabilités élémentaires de propagation dans la machine.

$P_{ab} = \frac{1}{8}$ probabilité d'absorption

$P_{nd} = \frac{4}{8} - \frac{1}{8} = \frac{3}{8}$ probabilité de non détection

$P_d = 1 - (P_{nd} + P_{ab}) = \frac{1}{2}$ probabilité de détection

La probabilité de succés de la propagation est donnée par l'expression suivante :

$$P_{SL} \# 1 - P_{EL} \# 1 - P_a (1 - P_d)^{L-1}$$

Le produit de la probabilité de localisation et de la probabilité de propagation permet d'obtenir la probabilité de détection, donc le taux de couverture de pannes τ.

Le tableau suivant donne, avec pour paramètre "a", le complément à 1 du taux de couverture qui permet le calcul direct du taux résiduel d'insécurité.

a	2	3	4	5	6	7	10
$1-\tau$	$3,8.10^{-1}$	$2,2.10^{-1}$	$1,2.10^{-1}$	$5,8.10^{-2}$	$3,0.10^{-2}$	$1,4.10^{-2}$	$4,0.10^{-3}$

La valeur de "a" trouvée pour la bascule D est a = 3, le graphe de la figure IV.9 en donne la topographie, cette modification est due à une coupure interne à la bascule. La défaillance a pour effet une absence de mémorisation de la variable donnée à l'état 1.

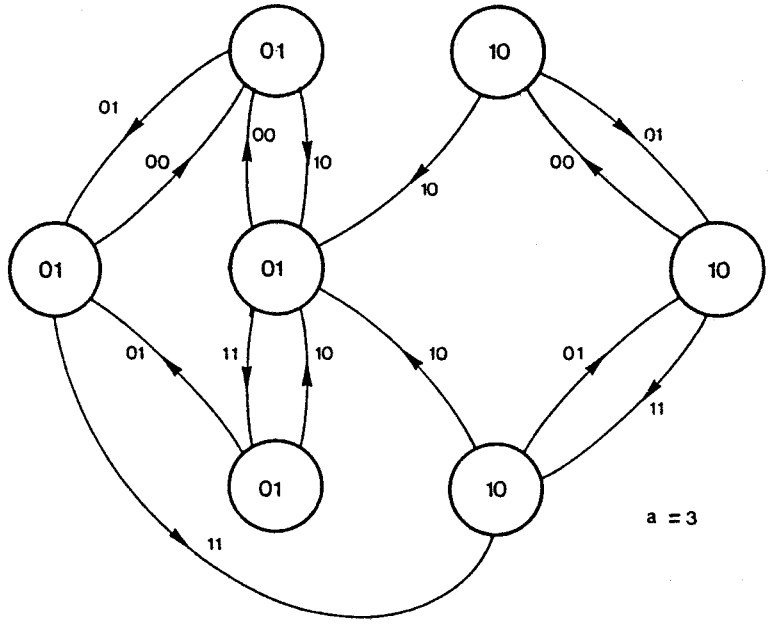


Fig.IV.9.

En prenant alors pour base de calcul cette valeur de "a" on trouve les valeurs suivantes pour chacune des quatre bascules.

$(1-\tau_1) = 1-\left[(1-2,1.10^{-1})\ (1-3,9.10^{-3})\right] = 2,2.10^{-1}$ bascule de mise en forme

$(1-\tau_2) = 2,2.10^{-1}$ bascule de prédynamisation m= 10

$(1-\tau_3) = 1,0.10^{-1}$ bascule de contrôle et dynamisation m= 8

$(1-\tau_4) = 1,0.10^{-1}$ bascule du générateur de fenêtre m= 8

Le taux résiduel d'insécurité est uniquement dû aux quatre bascules D dans la mesure où les fonctions combinatoires sont testées exhaustivement et, nous obtenons l'expression suivante :

$$\lambda_{St} = \lambda_{D} \sum_{i=1}^{i=4} (1 - C_i) = \lambda_{D} . 6,4 . 10^{-1}$$

avec λ_D, le taux de défaillance horaire d'une bascule D

La sécurité obtenue semble donc plus que modeste, en particulier, on remarque que ce chiffre est dû à la faible valeur des probabilités de localisation. En effet, les valeurs de m sont relativement importantes par rapport au nombre d'arcs total de la machine représentant les bascules.

Toutefois, cette faible valeur du taux de sécurité est à nuancer en fonction des hypothèses adoptées :

- le taux de couverture de pannes est obtenu pour la défaillance la plus difficile à détecter, cette valeur est alors à composer non pas avec le taux de défaillance global du circuit, mais avec un taux de défaillance relatif à la panne prise comme référence, c'est à dire celle modifiant trois arcs. Ce taux est nécessairement inférieur au précédent.

- nous avons également admis que toutes les défaillances non détectées sont dangereuses, alors qu'en réalité une partie de celles ci n'a aucune incidence sur la sécurité, certaines conduisent d'ailleurs à la prise de l'état de sécurité en cas de seconde panne.

Il est intéressant de voir de quelle manière ce chiffre peut être amélioré en particulier en renforçant la valeur de la probabilité de localisation.

IV.2.4.4) AMELIORATION DU TAUX DE COUVERTURE.

Une fois trouvé un arrangement optimal des fonctions logiques et un fonctionnement satisfaisant du contrôleur, il est difficile de reprendre la conception pour améliorer le test de quelques circuits.

Par contre, sous certaines conditions, il est possible d'augmenter le nombre d'arcs parcourus en fonctionnement normal.

Le graphe suivant, figure IV.10, est celui de la bascule de mise en forme avec en traits forts les arcs parcourus. On remarque que les arcs 13, 12, 10 et 11 peuvent être parcourus sans modifier la sortie donc sans perturber le fonctionnement de la structure.

Ainsi, à partir de l'état S_3 on parcourt l'arc 13 puis 12 et l'on revient dans l'état de départ, le même raisonnement peut être appliqué pour les arcs 11 et 10 à partir de l'état S_5.

Toutes les améliorations que nous allons faire vont utiliser ce principe.

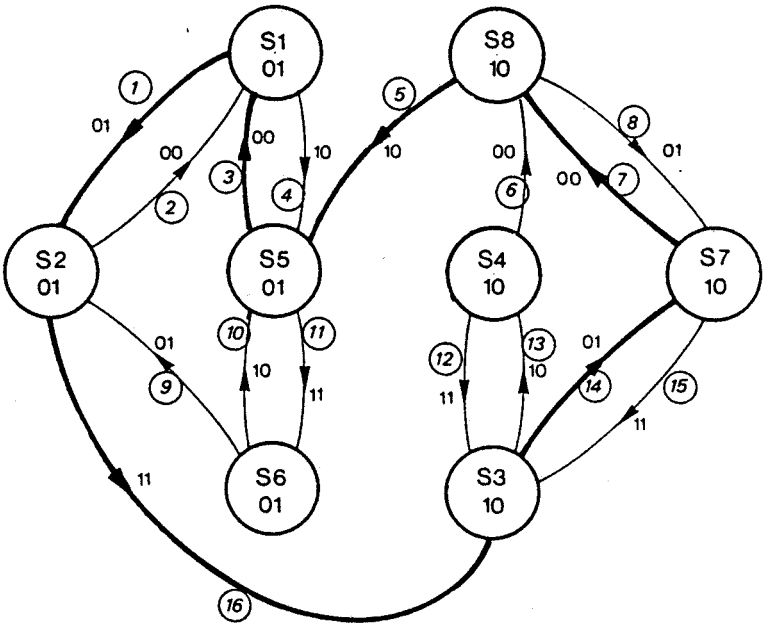


Fig.IV.10

Le gain est immédiat puisque m passe de 10 à 2, la longueur de la séquence est quant à elle augmentée de 12 soit L'= 18.

Et, on obtient pour la valeur de $1-\tau_4$:

$$(1-\tau_4) = 1-\left[(1-0)\ (1-9{,}5.10^{-3})\right] = 9{,}5.10^{-3}$$

L'amélioration est spectaculaire mais, il est nécessaire de regarder si les contraintes de supplément de test ne sont pas trop importantes.

Il faut :

- ne pas perturber le fonctionnement nominal du circuit testé.
- ne pas augmenter dans de grandes proportions les sources de pannes et ainsi trop diminuer la disponibilité du contrôleur.
- que le test soit sûr, une défaillance sur ces signaux supplémentaires doit également être détectée. En particulier, une défaillance provoquant la disparition de ce supplément de test sur une fonction ramène le taux résiduel à la valeur antérieure et l'objectif de sécurité n'est plus respecté.

Le schéma de la modification est donné figure IV.11 ainsi que les signaux de test supplémentaires.

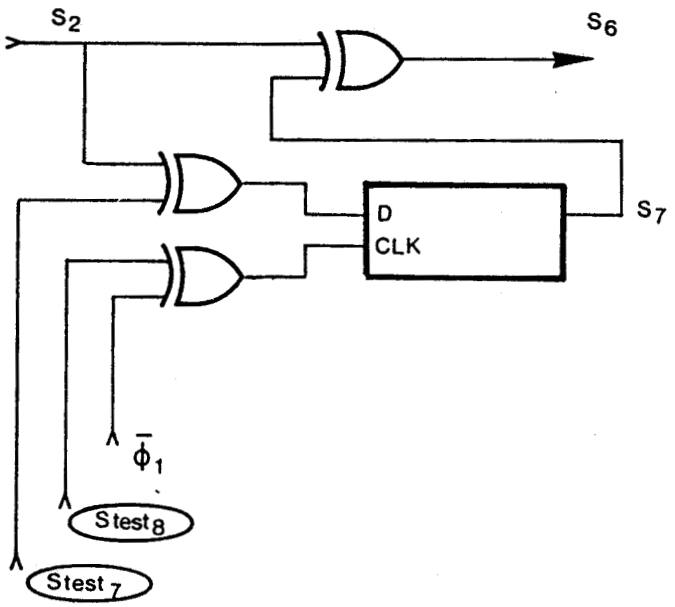
Les graphes de la figure IV.12 montrent la machine avant la modification de test et après ,les traits forts représentent toujours les arcs parcourus.

Les signaux issus de la base de temps servent par des combinaisons logiques simples à élaborer le supplément de test.

Au niveau des bascules, on introduit ceux ci par des portes de type OU exclusif.

En annexe 3 nous donnons les modifications nécessaires pour améliorer le taux de couverture de pannes des trois autres bascules.

Schéma complet du bloc de mise en forme



Stest$_7$ et Stest$_8$ sont les signaux supplémentaires permettant de compléter la séquence de test de la bascule D.

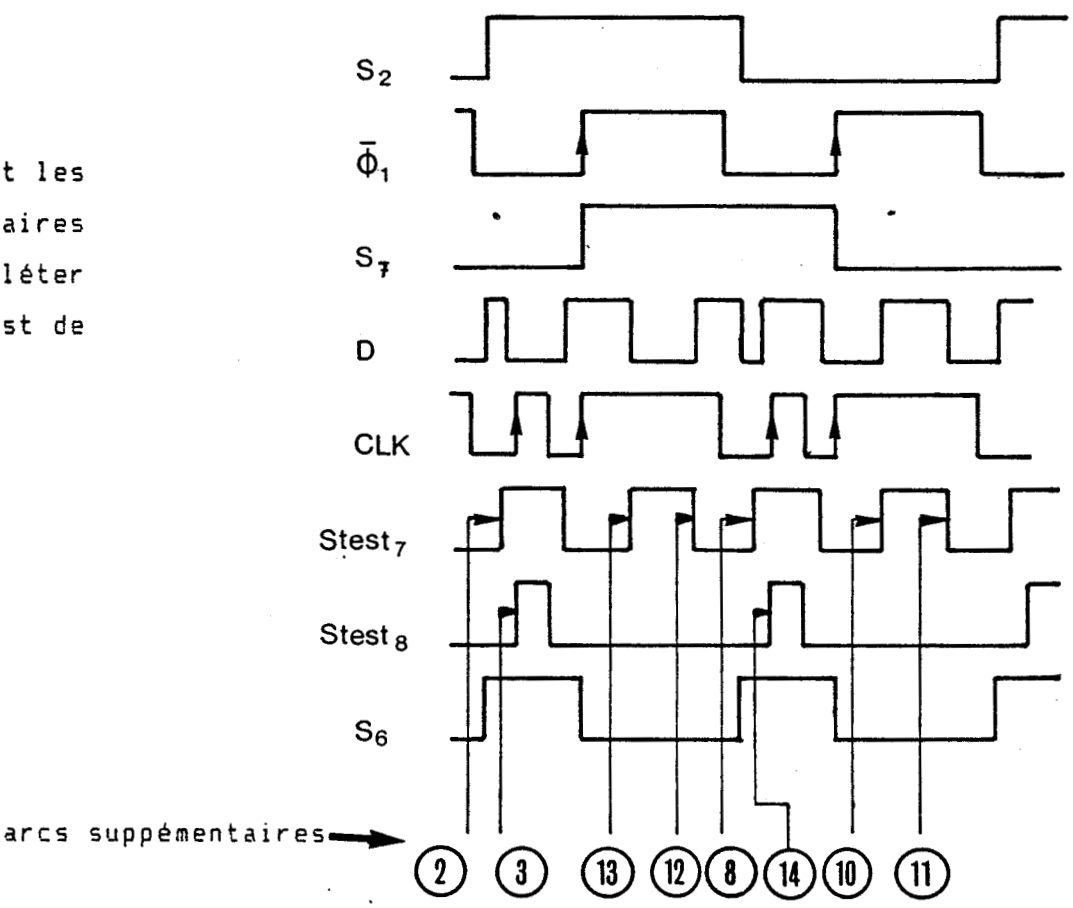
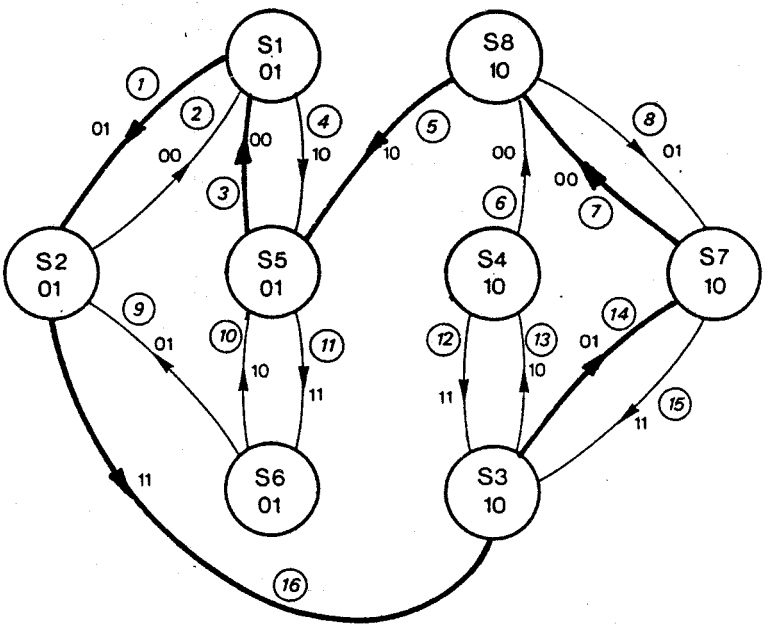


Fig.IV.11

Graphe de la bascule avant l'introduction de $Stest_t$ et $Stest_g$



Graphe de la bascule après, seuls les arcs 6 et 9 ne sont pas parcourus

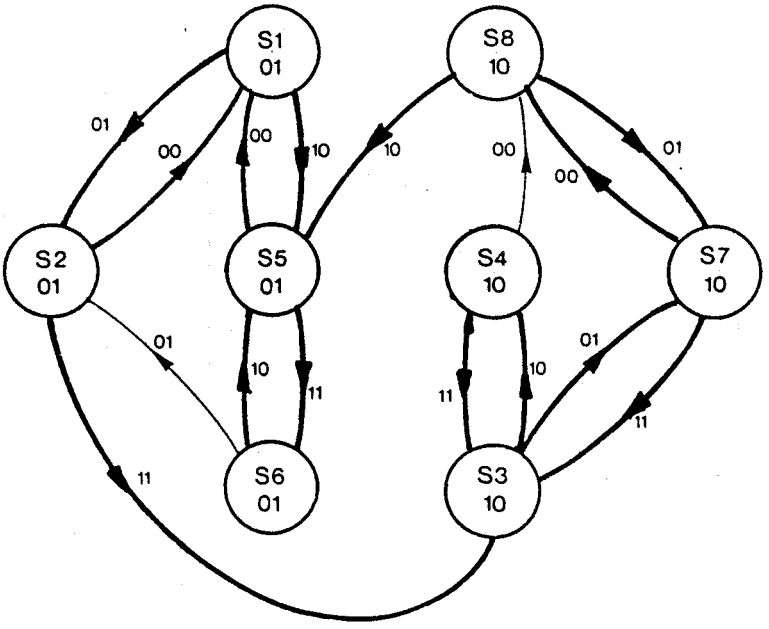


Fig.IV.12

Le tableau suivant permet la comparaison directe entre les résultats obtenus avant et après.

	Mise en forme	Pré dynamisation	Contrôle et dynamisation	Générateur de fenètre
arcs non parcourus avant	m = 10	m = 10	m = 8	m = 8
arcs non parcourus après	m = 2	m = 2	m = 1	m = 2
longueur séquence avant	L = 6	L = 6	L = 8	L = 194
longueur séquence après	L' = 18	L' = 14	L' = 16	L' = 190
$(1-\zeta)$ avant	$2,2.10^{-1}$	$2,2.10^{-1}$	$1,0.10^{-1}$	$1,0.10^{-1}$
$(1-\zeta)$ après	$9,5.10^{-7}$	$1,5.10^{-5}$	$3,8.10^{-6}$	$\neq 0$

On en tire la valeur du taux résiduel d'insécurité λ_{St}:

$$\lambda_{St} = \lambda_3 \sum_{i=1}^{i=4} (1-\zeta_i) = \lambda_3 \; 2,0.10^{-5}$$

soit une amélioration très sensible de l'ordre de 3.10^{-5}

IV.2.5) CONCLUSION.

Le contrôleur réalisé donne entière satisfaction, sa conception devait aboutir à l'origine à un produit en sécurité intrinsèque mais les problèmes de test en ont fait autrement.

Si les fonctions combinatoires, des portes à deux entrées, ne posent pas de problème de test, il en va autrement de leurs assemblages au sein des fonctions séquentielles.

Malgré une simplification extrême de celles ci (optimisation), les séquences de test ne peuvent être exhaustives. La maitrise des mécanismes de propagation des défauts demande des solutions particulières et certaines concessions, absorption de défauts mineurs, sont nécessaires.

Le taux résiduel d'insécurité est réduit de façon sensible en complétant artificiellement le volume de test que les signaux fonctionnels peuvent fournir.

On ne peut toutefois conclure que l'utilisation de fonctions logiques intégrées ne conduit jamais à des réalisations en sécurité intrinsèque, ponctuellement une solution peut être trouvée.

On note également que la logique dynamisée, ne se suffit pas en elle même et qu'en particulier, lorsqu'un signal statique est nécessaire, la conversion signal dynamisé - signal statique passe par l'emploi de circuits classiques de sécurité (composants discrets connus de manière exhaustive).

La logique dynamisée permet des traitements complexes que les circuits classiques réalisent mais avec des moyens disproportionnés, elle peut servir d'interface de sécurité performant entre le microprocesseur et un circuit de sécurité classique simple et sûr.

CONCLUSION GENERALE

L'objectif de notre étude était initialement d'aborder la conception de circuits contrôleurs de sécurité avec des fonctions logiques de faible intégration pour satisfaire des impératifs de performances fonctionnelles.

Après avoir défini la notion de circuit contrôleur dans le contexte de la mise en sécurité de processus puis, les problèmes spécifiques dûs à leur environnement, nous nous sommes attachés à lever les points particuliers liés à l'utilisation des circuits intégrés logiques.

Tout particulièrement, l'analyse des défaillances, l'influence de celles ci sur la fonction remplie, mais aussi la détection des mauvais fonctionnements ont débouché sur une vision probabiliste du test des circuits et donc, par là même, sur le concept de sécurité probabiliste à mettre en parallèle avec la sécurité intrinsèque.

Le taux de couverture de pannes qui est le rapport du nombre de pannes détectées sur le nombre de pannes possibles permet de quantifier la probabilité d'avoir un fonctionnement permissif dangereux lorsqu'une défaillance s'est produite.

La méthode de calcul décrite donne une limite inférieure dans la mesure où toute défaillance non détectée n'est pas nécessairement dangereuse mais aussi parce que ce calcul est mené en prenant pour hypothèse la détection de "la panne la plus difficile à détecter" ceci pour des circuits de type séquentiel.

La démarche pour mener à bien ce calcul consiste :

- à rechercher l'évènement très pénalisant pour le test (a)
- à déterminer la proportion de circuit non testée (m)
- à connaitre la longueur de la séquence de test (L)

La méthode a été mise à profit pour déterminer les points faibles d'une réalisation où par ailleurs un certain nombre de précautions avait déja été pris pour s'assurer de la sûreté de fonctionnnement.

Le taux résiduel d'insécurité a ainsi pu être réduit dans d'importantes proportions.

L'optimisme de départ qui consistait à dire que des fonctions logiques simples pouvaient encore mener à un contrôleur orienté sécurité intrinsèque a dû faire place à une réalité dans laquelle l'association de quelques portes réalisant un circuit séquentiel est d'une maitrise imparfaite en ligne. Ainsi, les règles précises d'assemblage pouvant constituer en quelque sorte des règles de l'art en matière de sécurité à base de circuits logiques intégrés n'ont pu être énoncées clairement.

REFERENCES BIBLIOGRAPHIQUES

1) A SCHWEITZER - J.P GERARDIN.
Méthodes permettant d'améliorer le niveau de sécurité des systèmes à logique programmée.
Revue ETI N 12 et N 13 . 1984.

2) Utilisation des microprocesseurs dans les applications ferrovières de sécurité.
Recherche bibliographique de l'IRT. Rapport MA 82_103 . Août 82

3) J.F WAKERLY.
Partially self checking circuits and their use in performing logical operations.
IEEE Trans on computers Vol C 23 . July 74

4) Processeur codé ou monoprocesseur de sécurité.
Journées " Sécurité des applications des automatismes numériques dans les transports ".
Villeneuve d'Ascq . 24 et 25 Octobre 84.

5) J.F DHALLUIN - J BAUDET.
Sécurité de fonctionnement des microprocesseurs par tests fonctionnels et observation temporelle.
Avancement des travaux de recherche pour l'année 1985.
GRRT. INRETS. USTL.

6) P VELAZCO.
Test comportemental des microprocesseurs.
Thèse de docteur ingénieur . INPG . Mars 82.

7) S MAGNIEZ.
Test fonctionnel de la partie opérative du microcontrôleur 8051
Rapport de DEA . USTL . Juin 84.

8) R GABILLARD - PH DELANGHE - F BARANOWSKI - J.F DHALLUIN.
Mise en sécurité d'un microprocesseurpar observation temporelle de son fonctionnement. Application à un réseau local de commande - contrôle de processus.
Journées " Sécurité des applications des automatismes numériques dans les transports ".
Villeneuve d'Ascq . 24 et 25 Octobre 84.

9) J.F DHALLUIN.
Commande - contrôle de processus en sécurité. application à la commande d'un ensemble de portes véhicule d'une rame de type VAL.
Thèse de docteur ingénieur . USTL . Décembre 83.

10) R GABILLARD.
Tentative de vérification de vraisemblance de l'affirmation de sécurité reposant sur le recours au concept de sécurité positive.
Note USTL . Diffusion MATRA EPALE . Mars 78.

11) F BARANOWSKI.
Utilisation des circuits cablés SSI pour la conception de fonctions en sécurité intrinsèque.
Rapport de DEA . USTL . Juillet 84.

12) F BOTQUIN.
Circuits intégrés logiques.
Techniques de l'ingénieur. Electronique Tome 2.
Articles E 1010 1011 1012 1013.

13) Pannes et erreurs dans les circuits intégrés logiques.
Recherche Bibliographique de l'IRT. Rapport MA 83_74 . Mai 83.

14) J.P VAUTRIN - M COLLIER.
Les fonctions dynamisme et autocontrôle et sécurité des systèmes logiques.
Revue Electronique Industrielle N 60 . Novembre 83.

15) J.E SMITH - PAKLIN LAM.
A theory of totally self checking system design.
IEEE . Trans on computers . Vol C 32 . September 83.

16) C ROBACH - G SAUCIER.
Le test logique des circuits intégrés.
Revue Onde Electrique N 58 . Décembre 78.

17) C MAGNIEZ.
Commande - contrôle par processus hiérarchisé
Thèse de 3 cycle . USTL . Décembre 85.

18) P THEVENOD FOSSE.
Test aléatoire de microprocesseurs 8 bits. Application au Motorola 6800.
Thèse d'état . Grenoble . Octobre 83.

19) R DAVID - P THEVENOD FOSSE.
Random testing of integrated circuits.
IEEE . Trans on instrumentation and measurement . Vol Im 30 October 80.

20) J.P HAYES.
On the properties of irredundant logic networks.
IEEE . Trans on computers . Vol C 25 . September 76.

21) D BIED CHARRETON.
Etude de faisabilité d'un microprocesseur autotestable.
Rapport INRETS CRESTA N°17 . Décembre 86.

22) D.A REYNOLDS - G METZE.
fault detection capabilities of alternating logic.
IEEE . Trans on computers . Vol C 27 . December 78.

23) J.F DHALLUIN - F BARANOWSKI.
Dispositif de contrôle de bon fonctionnement d'un microprocesseur mis en sécurité par observation temporelle.
Conception et étude de sécurité.
Rapport GRRT . Juin 85.

24) J.J MERCIER.
Contribution au test en ligne des circuits combinatoires.
Thèse de 3 cycle . USTL Montpellier . Juillet 74.

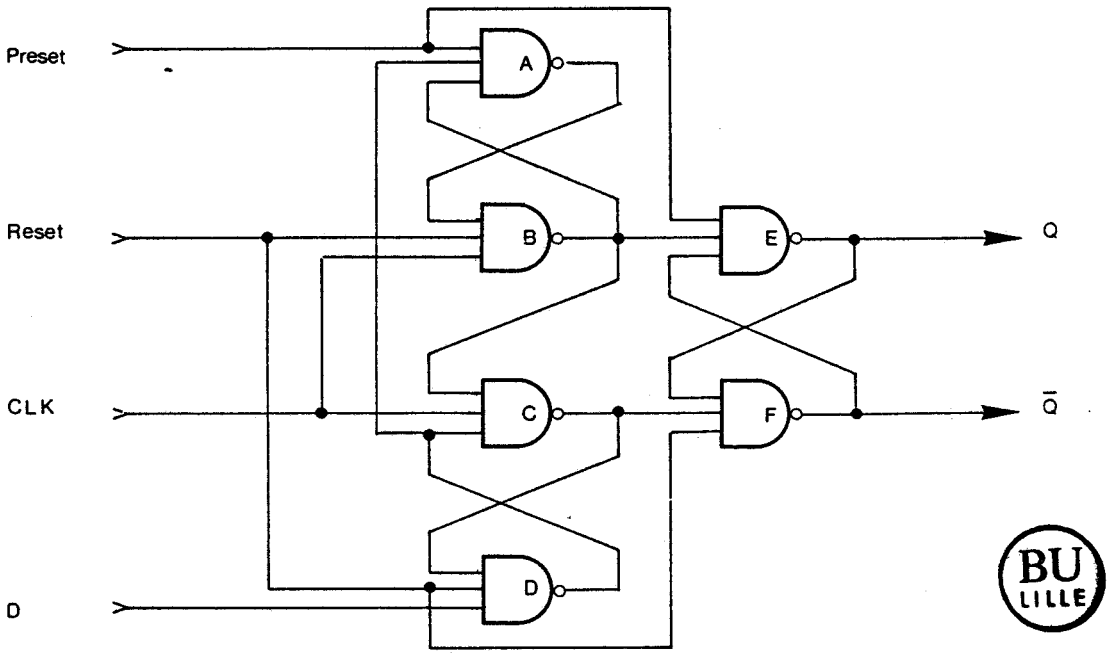
25) N GIAMBIASI.
Contribution au test en ligne des circuits séquentiels.
Thèse de 3 cycle . USTL Montpellier . Juillet 74.

ANNEXE 1

Le schéma d'une bascule D de type 7474 tel qu'on peut le trouver dans les manuels fournis par les constructeurs est donné à la suite.

Deux entrées normales D et CLK et deux sorties complémentaires Q et $\bar{Q}$.

Q recopie la valeur de D sur le front montant de CLK, cette valeur reste mémorisée tant qu'un autre front montant n'a pas appréhendé une nouvelle valeur de D.



Outre ses entrées classiques, la bascule possède deux autres entrées qui permettent de forcer l'état de la machine à tout instant.

Ces deux entrées compliquent singulièrement la table et le graphe décrivant le fonctionnement de la bascule.

On donne la table dans laquelle, on relève 24 états. Les quatre entrées permettent d'obtenir un ensemble de 4 x 24 = 96 arcs différents.

PRESET-RESET-CLK-D

Etats \ Entrées	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	10 00	10 01	10 10	11 11	11 00	11 01	11 10	11 11
S_1 1111 11	S_1	S_2	S_3		S_4				S_5							
S_2 1111 11	S_1	S_2		S_6		S_7				S_8						
S_3 1101 11	S_1		S_3	S_6			S_9				S_{10}					
S_4 1111 10	S_1				S_4	S_7	S_9						S_{11}			
S_5 0111 01	S_1								S_5	S_8	S_{10}		S_{13}			
S_6 1101 11		S_2	S_3	S_6				S_{12}				S_{14}				
S_7 1110 10		S_2			S_4	S_7		S_{12}						S_{15}		
S_8 0111 01		S_2							S_5	S_8		S_{14}		S_{16}		
S_9 1011 10			S_3		S_4		S_9	S_{12}							S_{17}	
S_{10} 0101 01			S_3						S_5		S_{10}	S_{14}			S_{18}	
S_{11} 0111 10					S_4				S_{19}				S_{11}	S_{15}	S_{17}	
S_{12} 1010 10				S_6		S_7	S_9	S_{12}								S_{20}
S_{13} 0111 01					S_{21}				S_5				S_{13}	S_{16}	S_{18}	
S_{14} 0101 01				S_6						S_8	S_{10}	S_{14}				S_{22}
S_{15} 1110 10						S_7			S_{19}				S_{11}	S_{15}		S_{20}
S_{16} 1110 01						S_{23}			S_5				S_{13}	S_{16}		S_{20}
S_{17} 1011 10							S_9				S_{10}		S_{11}		S_{17}	S_{20}
S_{18} 0101 01							S_9				S_{10}		S_{13}		S_{18}	S_{22}
S_{19} 0111 10	S_1								S_{19}	S_{24}	S_{10}		S_{11}			
S_{20} 1010 10								S_{12}				S_{14}		S_{15}	S_{17}	S_{20}
S_{21} 1111 01	S_1				S_{21}	S_{23}	S_9						S_{13}			
S_{22} 1010 01								S_{12}				S_{14}		S_{16}	S_{18}	S_{22}
S_{23} 1110 01		S_2			S_{21}	S_{23}		S_{12}						S_{16}		
S_{24} 0111 10		S_2							S_{19}	S_{24}		S_{14}		S_{15}		

ABCD EF

TABLE D'EVOLUTION D'UNE BASCULE D A QUATRE ENTREES

ANNEXE 2

La bascule D représentée ici est optimisée, elle ne comporte que le nombre d'entrées nécessaire à un fonctionnement mémoire.

En particulier, les deux entrées de positionnement de l'état de la machine sont absentes (RESET PRESET).

De ce fait, le nombre d'états est limité à 8 et le nombre d'arcs possibles à 16.

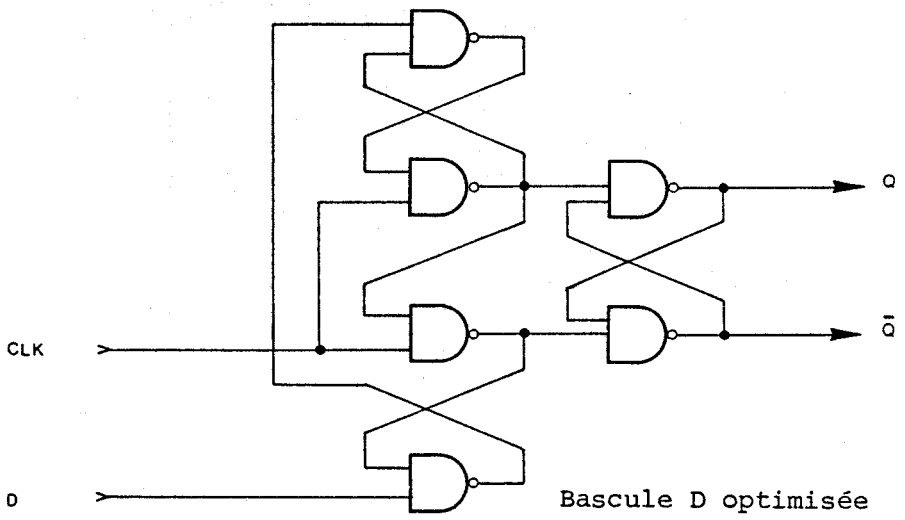
Le graphe fait apparaitre une certaine symétrie.

- 4 états dont la sortie vaut 01
- 4 états dont la sortie vaut 10
- 1 arc de passage de la sortie 01 vers la sortie 10
- 1 arc de passage de la sortie 10 vers la sortie 01

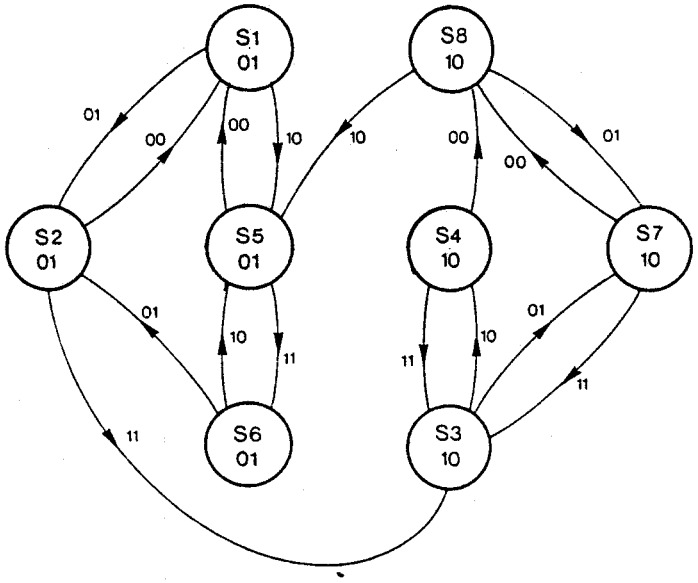
Outre l'inconvénient de positionner l'état de la machine par les entrées normales D et CLK (séquence de positionnement) au lieu de RESET PRESET (positionnement immédiat), les séquences de test sont beaucoup plus réduites et favorisent l'utilisation de cette machine dans une réalisation de sécurité.

La page suivante donne :

- le schéma logique
- le graphe
- la table d'évolution.



Bascule D optimisée



Etats \ Entrées	00	01	11	10
S_1 01. 11	(S1) 01	S_2		S_5
S_2 11. 10.	S_1	(S2) 01	S_3	
S_3 10. 10		S_8	(S3) 10	S_4
S_4 10. 11	S_8		S_3	(S4) 10
S_5 01. 01	S_1		S_6	(S5) 01
S_6 01. 01		S_2	(S6) 01	S_5
S_7 11. 10	S_8	(S7) 10	S_3	
S_8 01. 11	(S8) 10	S_7		S_5

ANNEXE 3

Le but du complément de test est de faire parcourir à la machine des arcs et donc des chemins que les signaux de fonctionnement normaux ne lui font pas emprunter.

Pour cela, on peut mettre à profit le fait que certains états possèdent les mêmes sorties. Il est alors possible d'évoluer dans le graphe sans perturber extérieurement les sorties donc le contrôleur sauf, si la machine est affectée par une défaillance auquel cas on détecte celle ci.

Pour les bascules des trois blocs restants :

- bloc de prédynamisation
- bloc de contrôle et dynamisation
- bloc générateur de fenètre

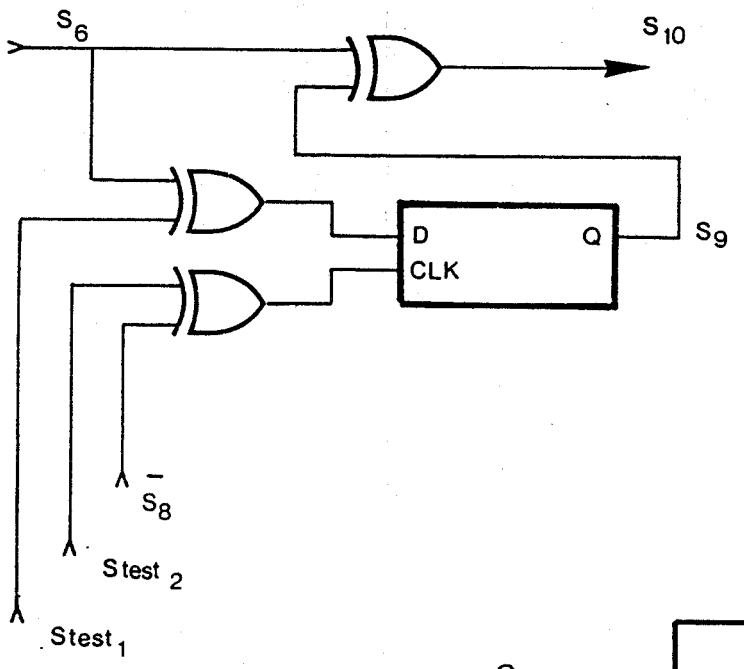
on donne le schéma et les signaux utilisés ainsi que les graphes avant et après supplément de test.

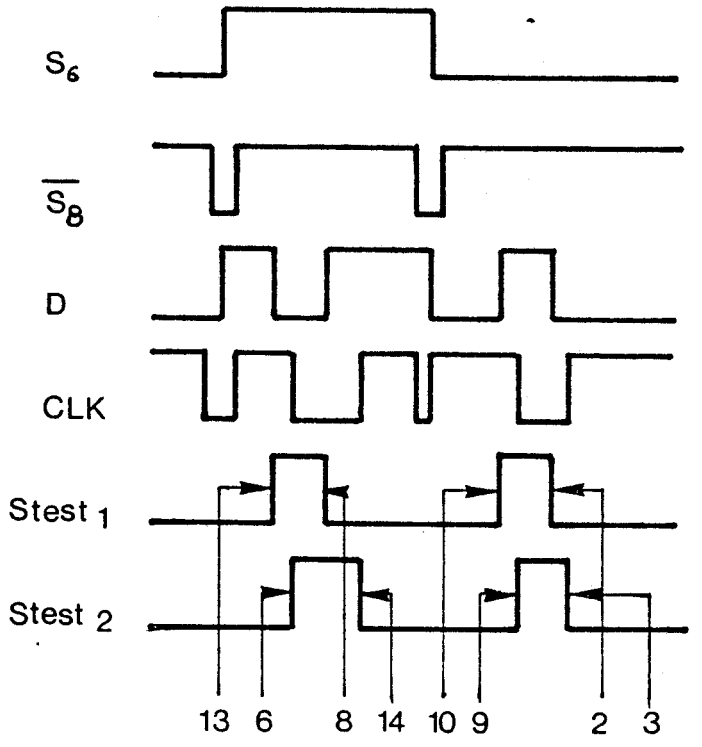
Le gain en couverture de test est important du fait que :

- les graphes sont mieux "explorés", la probabilité de non localisation diminue avec l'importance de l'exploration.
- les séquences de test sont allongées, la probabilité de propagation est meilleure.

Les défauts qui pourraient apparaitre sur ces suppléments de test sont propagés en venant altérer les signaux de fonctionnement normaux de la bascule.

BLOC DE PREDYNAMISATION





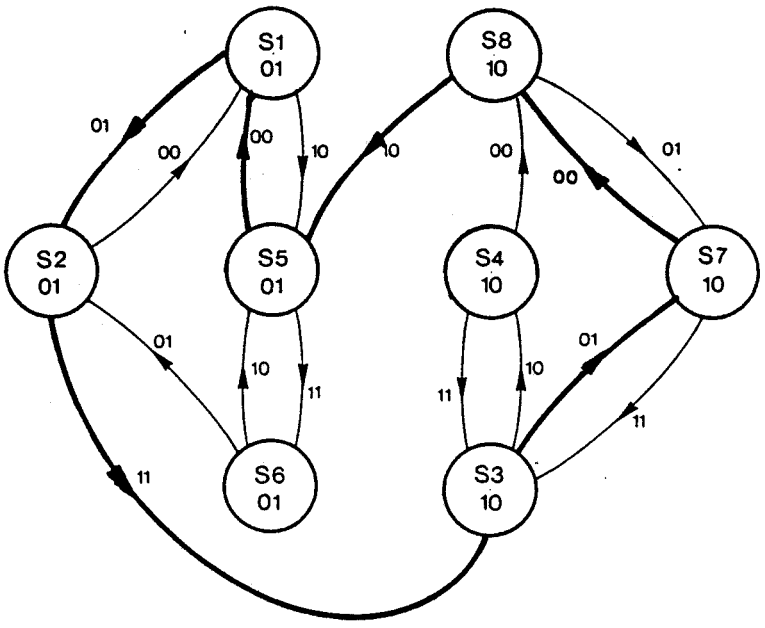
arcs parcourus par le supplément →

Tel qu'il a été conçu au départ, ce bloc reçoit les signaux S_6 et $\overline{S_8}$ d'où une séquence de longueur L = 6, le nombre d'arcs non parcourus est très important m = 10 pour un total de 16 à parcourir.

On ajoute $Stest_1$ et $Stest_2$ pour compléter le test.

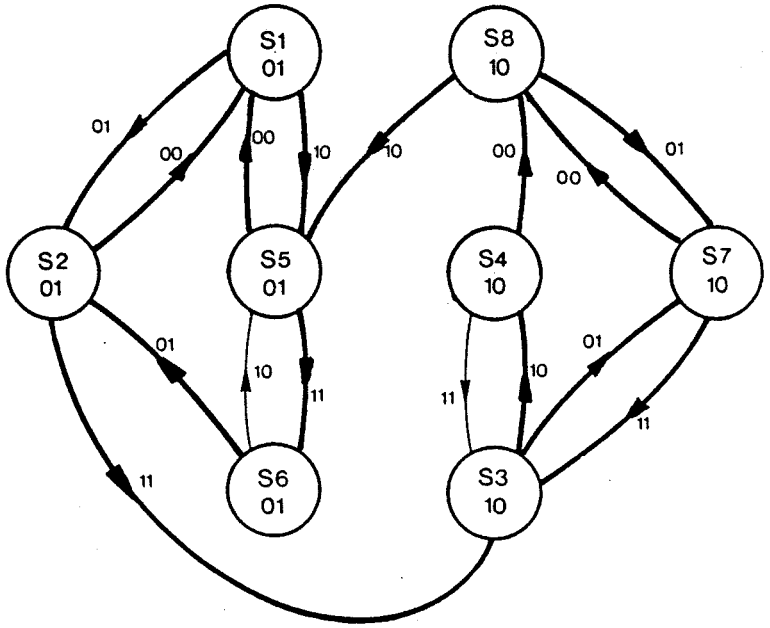
Graphe sans le supplément

$m = 10$ et $(1-\zeta_2) = 2{,}2.10^{-1}$

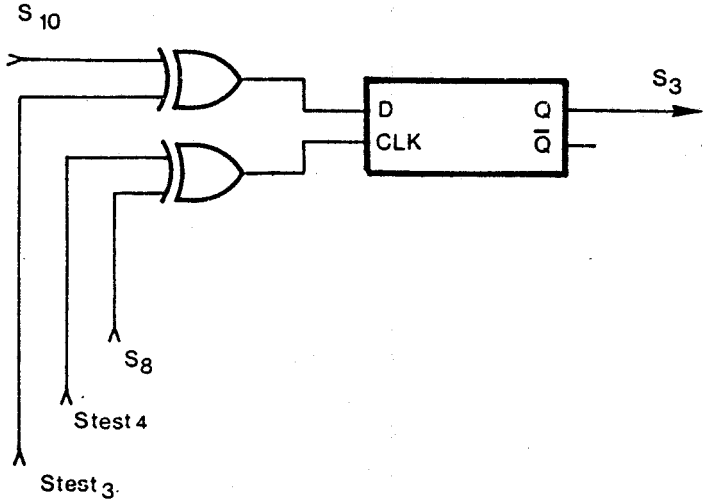


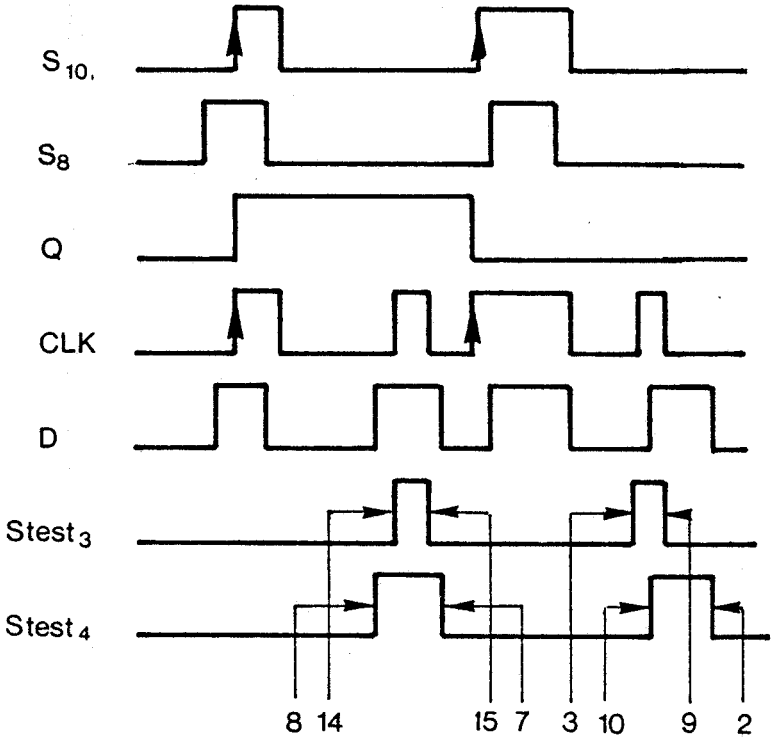
Graphe avec supplément

$m = 2$ et $(1-\zeta_2) = 1{,}5.10^{-2}$



BLOC DE CONTROLE ET DYNAMISATION

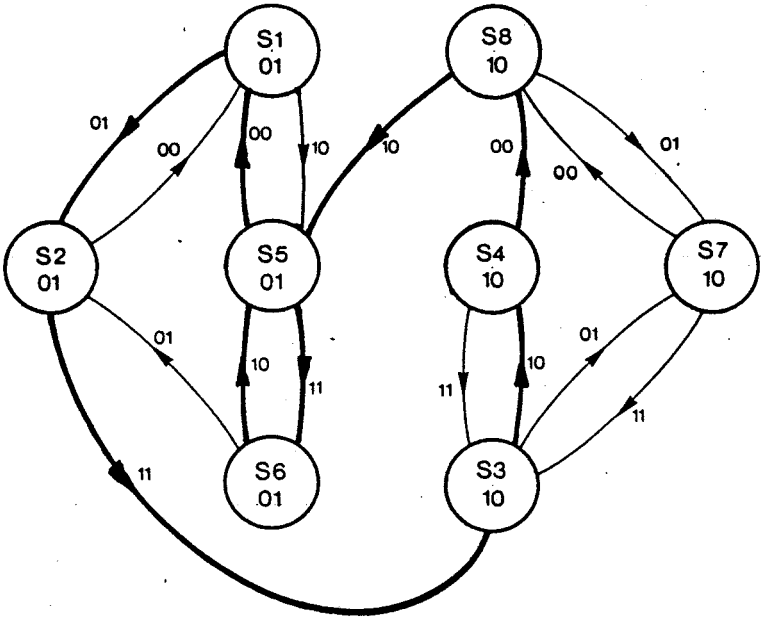




Normalement testé par S_8 et S_{10}, la longueur du test est L = 8 et le nombre m = 8 soit la moitié des arcs de la machine non parcourus.

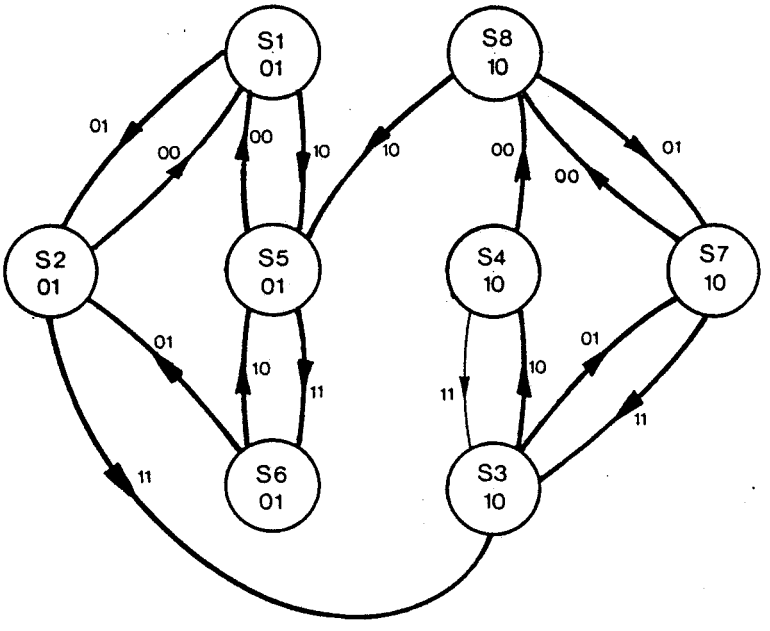
Graphe sans le supplément

$m = 8$ et $(1-\zeta_3) = 1,0.10^{-1}$

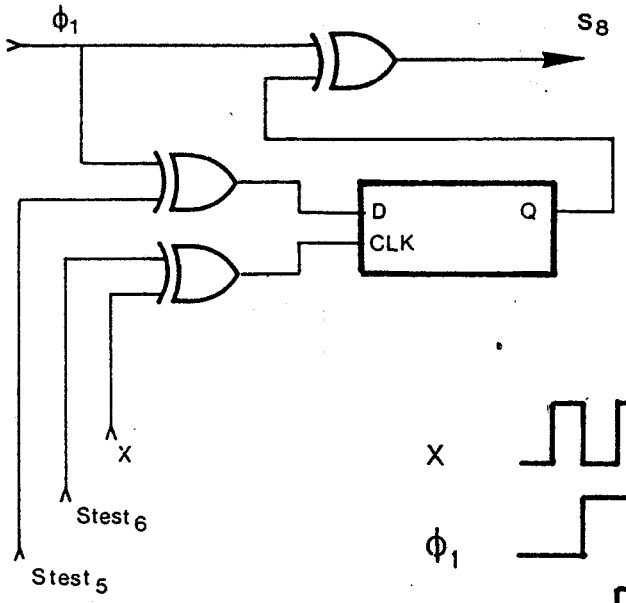


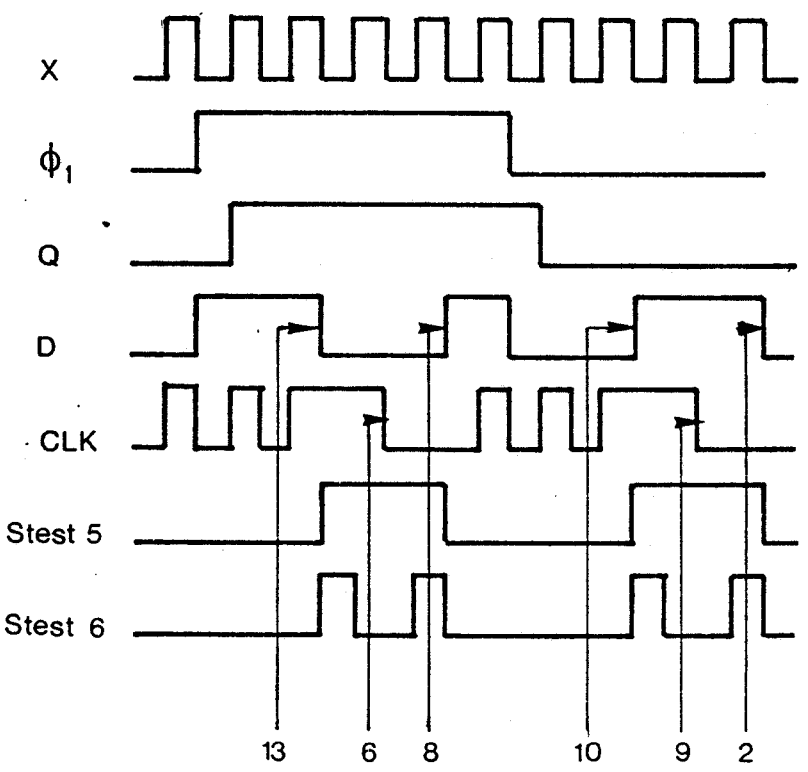
Graphe avec supplément

$m = 1$ et $(1-\zeta_3) = 3,8.10^{-6}$ avec $L = 16$



BLOC GENERATEUR DE FENETRE



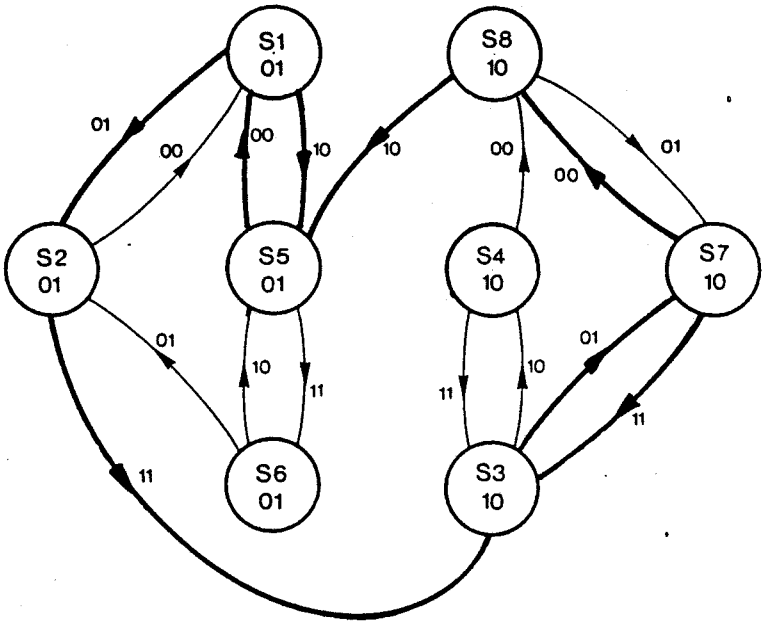


Les signaux servant au test de ce bloc sont ϕ_1 et X, la longueur du test est L = 194 (période de X = 2µS), le nombre m = 8.

L'introduction de $Stest_6$, $Stest_5$ permet d'obtenir m = 2, L = 190

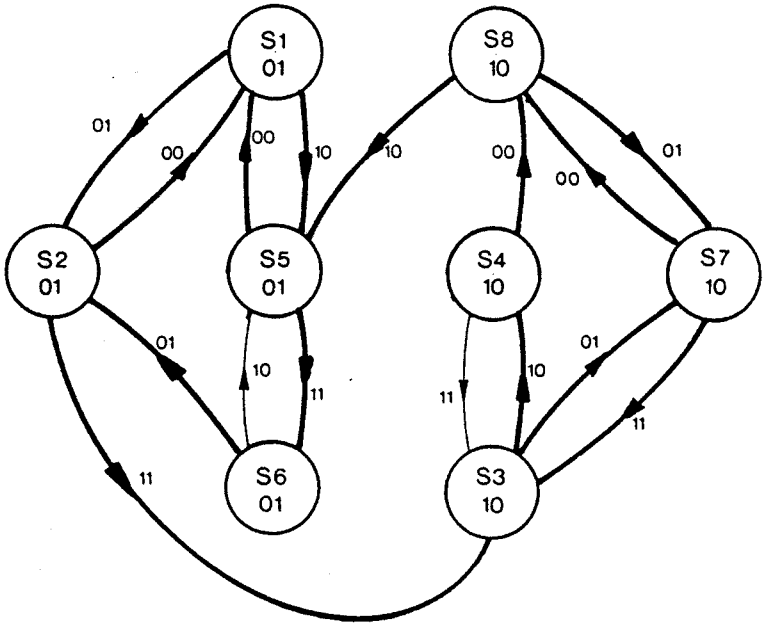
Graphe sans le supplément

$m = 8$ et $(1-\zeta_4) = 1,0.10^{-4}$



Graphe avec le supplément

$m = 2$ et $(1-\zeta_4) \neq 0$



ANNEXE 4

La structure de contrôle complète est donnée aux pages suivantes. On reconnait les quatre sous ensembles principaux connectés à la base de temps et le retour par la cellule de retard.

Trois signaux alimentent le contrôleur :

- S_1 qui est le signal issu du quartz du microprocesseur sous contrôle (12 MHz) et qui permet d'avoir un fonctionnement synchrone de celui ci.
- S_2 , le signal équitemps à contrôler et dont les caractéristiques ont été décrites dans le cahier des charges.
- Sinit, le signal généré par le microprocesseur avant toute autre application (gestion du processus) et qui est destiné à configurer correctement le contrôleur à la mise sous tension.

La séquence complète de configuration est donnée plus loin, elle utilise le signal Sinit pour positionner l'état de trois des quatre bascules, celle de contrôle et dynamisation étant positionnée par une impulsion de S_2.

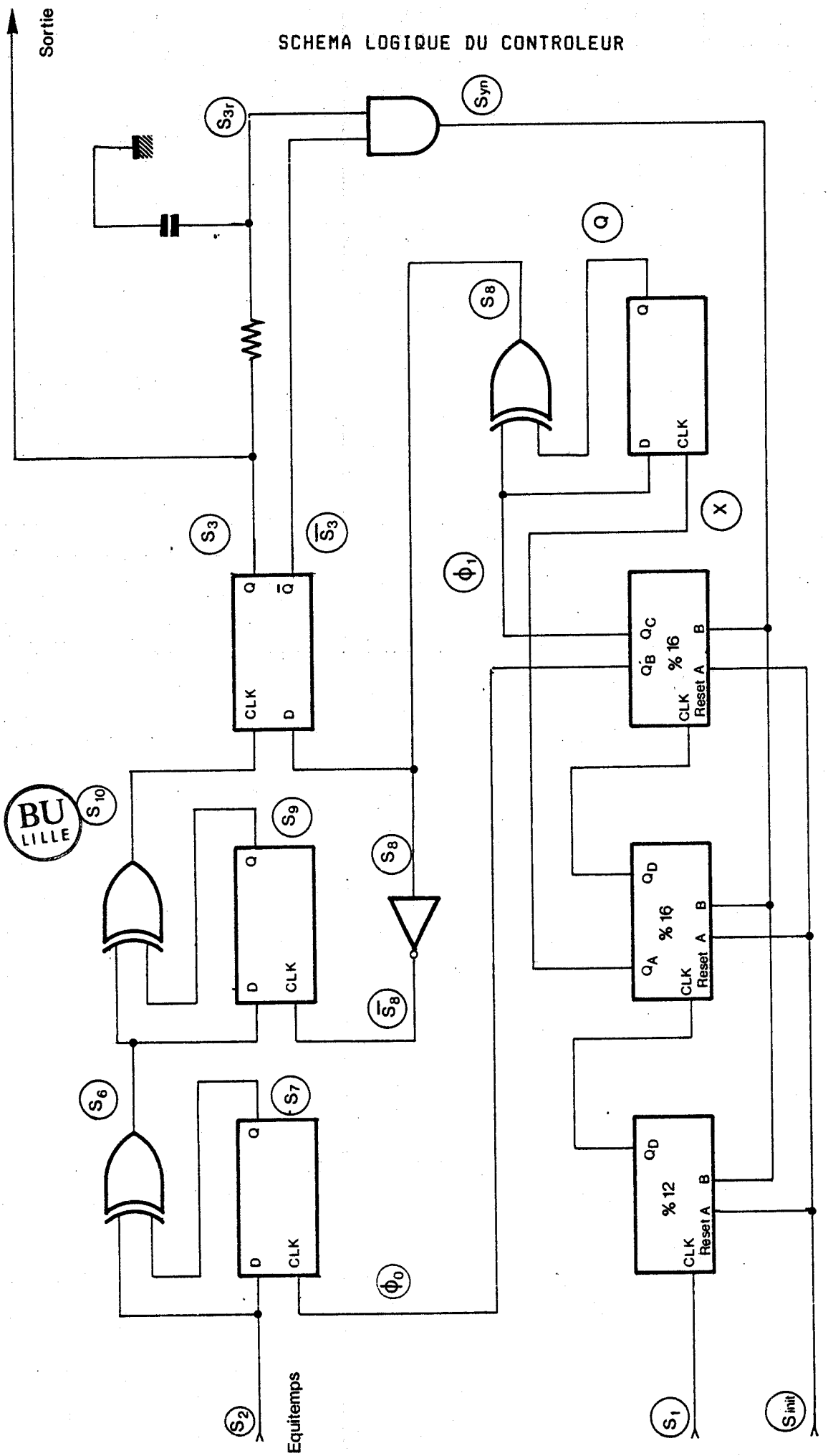
La phase d'initialisation est nécessaire pour :

- positionner l'état des bascules qui après une mise sous tension non pas d'état préférentiel or, il faut que leurs sorties soient précisément connues pour "amorcer" le contrôle.
- "caler" le premier front du signal équitemps avec la première fenètre délivrée par le contrôleur.

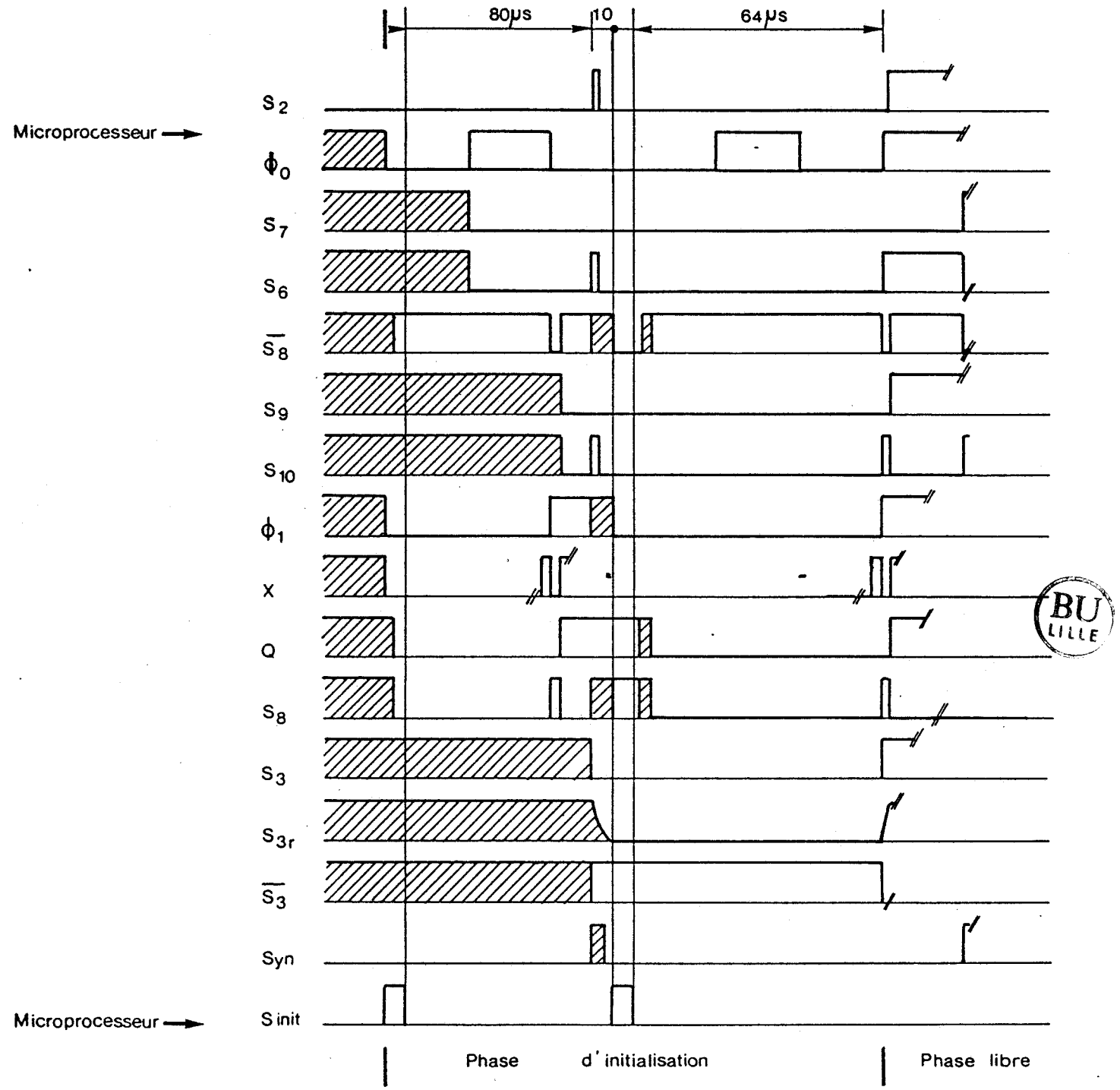
Sinit n'a plus alors aucun rôle autre que perturbateur et détectable si son état n'est pas stable après la phase d'initialisation.

La structure de contrôle est totalement autonome après l'initialisation.

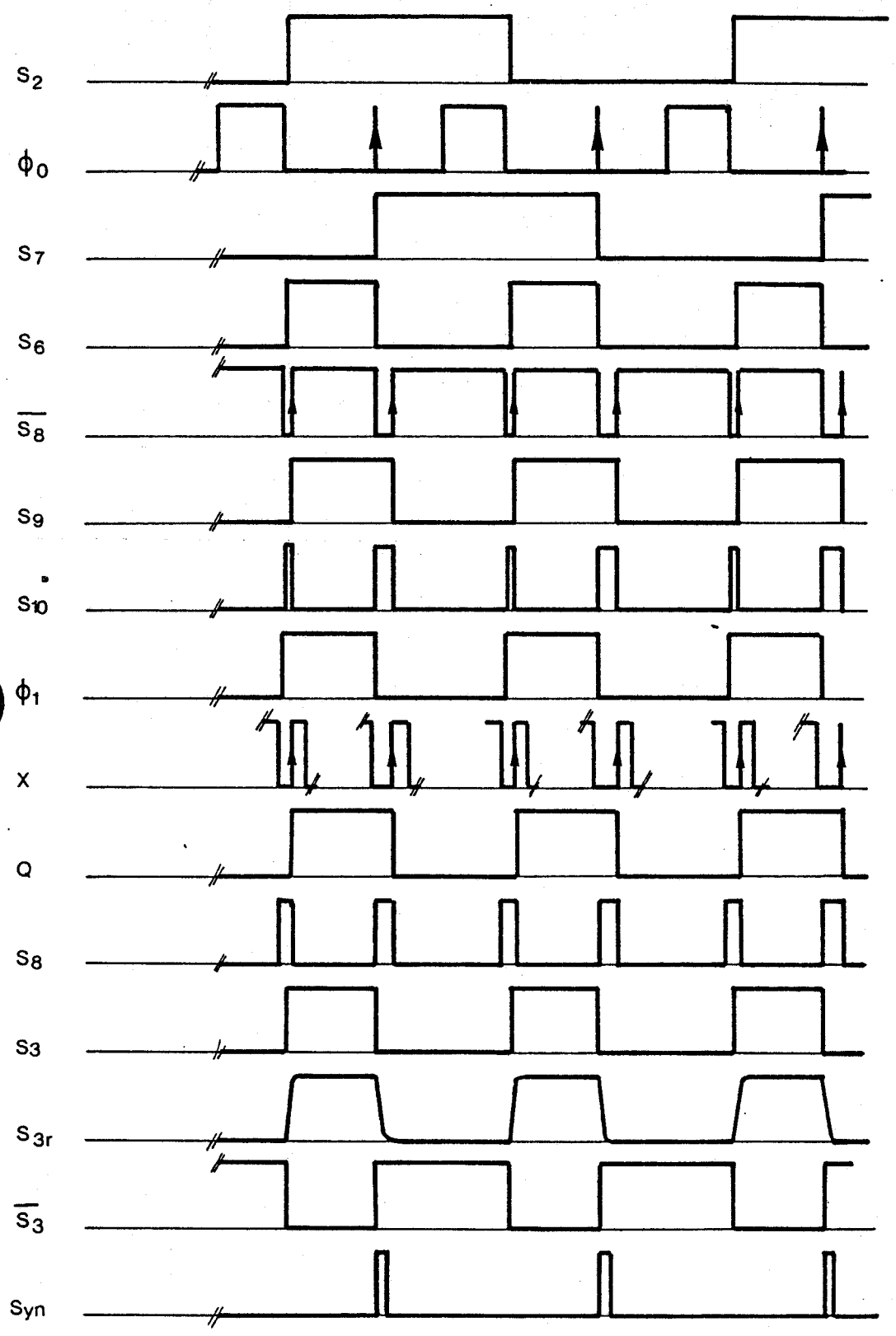
SCHEMA LOGIQUE DU CONTROLEUR



SEQUENCE D'INITIALISATION DU CONTROLEUR



SEQUENCE DE FONCTIONNEMENT AUTONOME EN L'ABSENCE DE DEFAILLANCE



ANNEXE 5

La logique alternative est très intéressante pour la conception de contrôleur de sécurité manipulant des informations non prévisibles.

Les comparateurs et les voteurs sont typiquement des contrôleurs qui trouvent une solution satisfaisante par cette méthode.

La logique alternative confère aux grandeurs manipulées le caractère déterministe nécessaire au suivi sécuritaire de l'information.

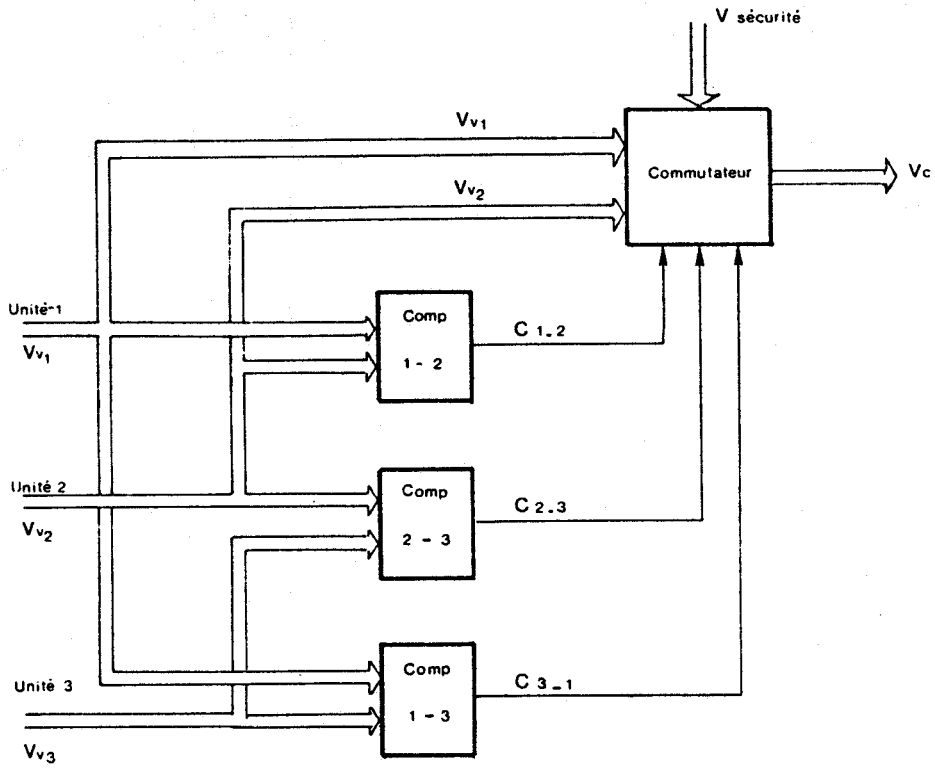
Après avoir défini une architecture satisfaisant à la tâche fonctionnelle souhaitée et élaboré l'ensemble des équations logiques résolvant localement les problèmes, il suffit de transposer chaque fonction en fonction alternative possédant donc les propriétés de dualité.

Nous présentons ici succintement, l'exemple d'un contrôleur du type voteur 2 parmi 3.

Trois unités en redondance sont comparées, la sortie est validée lorsqu'au minimum deux unités délivrent une sortie identique.

L'unité défaillante est automatiquement exclue.

Le vecteur de sécurité est délivré lorsque plus de deux unités divergent.



Les trois unités redondantes sont comparées deux à deux et bit à bit par des comparateurs délivrant une information 1 s'il y a accord entre chaque bit de même poids de chaque unité.

Comparateur bit à bit en logique alternative

Φ	a_n	b_n	C_n
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0

$C = \Phi \overline{(a \oplus b)} + \overline{\Phi} (a \oplus b)$

C = (1,0) concordance (info 1)

C = (0,1) divergence (info 0)

C = (0,0) défaillance dans le comparateur ou les lignes qui l'alimentent

C = (1,1) idem

Chaque comparateur est composé de n cellules (n = dimension des vecteurs d'entrée à contrôler), les sorties de celles ci sont regroupées en un ET en logique alternative.

On dispose ainsi de 3 sorties unifilaires C12, C23, C31 dont les valeurs alternatives sont :

0 = divergence entre un ou plusieurs bits

1 = les bits sont tous identiques deux à deux

Ces trois sorties pilotent alors un commutateur qui selon les valeurs de C12, C23, C31 délivre :

- le vecteur de sortie de l'unité 1
  si l'une des deux unités 2 ou 3 est défaillante
  (une seule à la fois)

- le vecteur de sortie de l'unité 2
  si l'une des deux unités 1 ou 3 est défaillante

- le vecteur de sortie sécuritaire $\overrightarrow{V_s}$
  si plus d'une unité est défaillante

Le tableau de fonctionnement de ce commutateur est le suivant :

C12	C23	C31	V_c
0	0	0	V_s
0	0	1	V_{u_1}
0	1	0	V_{u_2}
0	1	1	V_s
1	0	0	V_{u_1}
1	0	1	V_s
1	1	0	V_s
1	1	1	V_{u_2}

Le commutateur peut également être décomposé en une partie commande et une partie commutation.

PARTIE COMMANDE

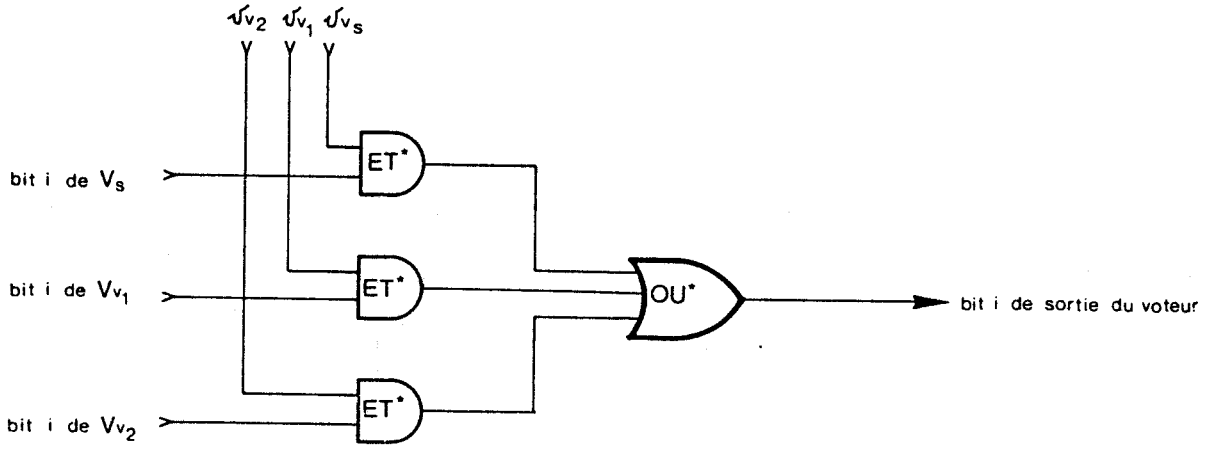
Elle consiste à élaborer trois signaux de commande en fonction des sorties à commuter. Le tableau suivant décrit cette partie.

C12	C23	C31	v	v	v
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	1

V_s , V_{w_1}, V_{w_2} commandent la commutation de V_s , V_{v_1}, V_{v_2}

PARTIE COMMUTATION

La partie commutation est un motif répété n fois.



RESUME

Le respect de sévères contraintes de sécurité pour les processus gérés par microprocesseurs passe entre autre par le contrôle du bon fonctionnnement de ces derniers. Ainsi, le contrôleur, adapté à une méthode de détection de défaillances, est une circuiterie logique indépendante qui en dernier ressort valide ou annule les commandes élaborées par le ou les microprocesseurs.

Afin de garantir à cette tache une certaine crédibilité vis à vis de la sécurité, le dispositif doit être autotesté : en présence d'une défaillance qui l'affecte et risque d'altérer son travail, il inhibe par excès les commandes des microprocesseurs.

Pour l'autotest, l'approche par le concept de logique dynamisée permet d'obtenir une excitation maximale et un test permanent des circuits utilisés. L'analyse des contraintes d'un test fonctionnel exhaustif pour les circuits combinatoires et séquentiels montre que l'investigation en ligne ne peut souvent qu'être partielle, on est alors amené à chiffrer de manière probabiliste le taux de couverture de pannes obtenu pour chaque circuit en fonction de la longueur et du contenu des séquences de test.

La fin de ce rapport décrit deux moyens de mise en oeuvre de la logique dynamisée et donne, pour un contrôleur concrèt, le calcul du taux de couverture de chaque élément le constituant et l'insécurité résiduelle finale.

MOTS-CLES :

Sécurité - Circuits autotestés - Contrôleurs - Circuits SSI MSI
Détection de pannes - Taux de couverture - Probabilité de détection.