*N° d'ordre*

Année : 2013 / 2014

جامعة عبد المالك السعدي
Université Abdelmalek Essaadi

# UNIVERSITE ABDELMALEK ESSAADI

FACULTE DES SCIENCES et TECHNIQUES, TANGER

**Centre d'Etudes Doctorales : « Sciences et Techniques de l'Ingénieur »**

**Formation Doctorale : « Sciences et Techniques de l'Ingénieur »**

## THESE DE DOCTORAT

Présentée

pour l'obtention du

DOCTORAT EN SCIENCES ET TECHNIQUES DE L'INGENIEUR

Par :

**Mohammed BSISS**

**Discipline : Electronique, Electrotechnique, Automatique (EEA)**

**Spécialité : Sécurité dans les systèmes embarqués**

Titre de la Thèse **: Sûreté de fonctionnement d'un système d'inférence floue avec une architecture redondante un parmi deux avec diagnostic (1oo2D) à base de FPGA**

*Soutenue le 26 décembre 2013 devant le jury :*

| | | |
|---|---|---|
| Pr. Mohammed ADDOU | Doyen FST Tanger | Président |
| Pr. Fouad LAHJOMRI | ENSAT Tanger | Rapporteur |
| Pr. Zine El Abidine ALAOUI ISMAILI | ENSIAS Rabat | Rapporteur |
| Pr. Mohammed BOUHORMA | FST Tanger | Rapporteur |
| Pr. Oualid KAMACH | ENSAT Tanger | Examinateur |
| Pr. M'hamed AIT KBIR | FST Tanger | Examinateur |
| Pr. Benaissa AMAMI | FST Tanger | Directeur de thèse |

Structure de recherche accréditée d'accueil :

Laboratoire Informatique, Systèmes et Télécommunications (LIST) de la FST de Tanger

# Sûreté de fonctionnement d’un système d’inférence floue avec une architecture redondante un parmi deux avec diagnostic (1oo2D) à base de FPGA

# Table des matières

# Table des figures

# Table des tableaux

# Nomenclature

| | |
|---|---|
| ADC | Convertisseur analogique-numérique |
| AMDEC | Analyse des modes de défaillance, de leurs effets et de leur criticité |
| ASIC | *Application specific integrated circuit*, circuits intégrés spécifiques à une application |
| ATPG | *Automatic test pattern generator* |
| BDF | Blocs-diagrammes de fiabilité |
| CEI | Commission électrotechnique internationale |
| CF | Fautes de couplages |
| CCF | Défaillance de cause commune |
| CLB | Bloc logique programmable |
| CRC | Contrôle de redondance cycle |
| DAC | Convertisseur numérique-analogique |
| DC | Facteur de couverture du diagnostic |
| DSP | Processeurs de signaux numériques |
| E/E/PE | Systèmes électriques, électroniques et programmables E/E/EP |
| FPGA | *Field programmable gate array*, réseau de portes programmable |
| IOB | *Input-output blocs*, les blocs d'entrées et sorties |
| ISA, | *Instrumentation, systems and automation society* |
| ISO | *International standardardisation organization*, organisme international de normalisation |
| MIF | Moteur d'inférence floue |
| MTBF | *Mean time between failures*, durée moyenne entre deux défaillances successives |
| MTTF | *Mean time to failure*, durée moyenne de bon fonctionnement avant la première défaillance |
| MTTR | *Mean time to repair*, durée moyenne de réparation |
| MooN | Pour caractériser l'architecture d'un système la convention MooN sera utilisée, ce qui signifie que M canaux sur les N canaux que compte le système doivent fonctionner correctement pour que la fonction de sécurité soit exécutée |
| PES | *Programmable electronic system*, système électronique programmable |
| PFD | *Probability of failure on demand*, probabilité de défaillance à la demande |
| PFDavg | *Average probability of failure on demand*, la probabilité moyenne de défaillance sur demande |
| PFH | *Probability of a dangerous failure per hour*, probabilité de défaillance dangereuse par heure |
| PID | Contrôleur proportionnel intégral dérivé |

| | |
|---|---|
| PLr | *Required performance level,* le niveau de performance |
| PSF | Fautes de voisinage Pattern Sensitive |
| RDA | Registre au décalage des adresses |
| RDD | Registre de décalage des données |
| RTL | *Register-transfer level,* description au niveau RTL |
| RRF | Facteur de réduction du risque |
| SAF | Fautes de blocage *stuck-at fault* |
| SAS | Système automatisé de sécurité |
| SIF, | *Safety instrumented function*, fonction instrumentée de sécurité |
| MIFS | Moteur d'inférence floue à sécurité |
| SIL | *Safety integrity level*, niveaux d'intégrité de sécurité |
| SIS | Systèmes instrumentés de sécurité |
| SRAM | Mémoire statique |
| TF | Fautes de transition |
| TTM | *Time to market,* temps de mise sur le marché |
| VHDL | *Very high speed integrated circuit hardware description language*, langage de description du matériel |

# Remerciements

Les travaux présentés dans cette thèse ont été réalisés au sein du Laboratoire d'Informatique Systèmes et Télécommunications (LIST), à la Faculté des Sciences et Techniques de Tanger (FST) de l'Université Abdelmalek Essaadi.

Je voudrais remercier Madame le Professeur, Amina AZMANI et Monsieur le Professeur Mohammed ADDOU, doyens de la Faculté des Sciences et Techniques de Tanger pour m'avoir accueilli à la FST et pour toute l'aide qu'ils m'ont apporté pour que cette thèse se fasse dans les meilleures conditions.

Je tiens tout particulièrement à exprimer ma plus profonde gratitude à Monsieur le Professeur Benaissa AMAMI, mon Directeur de recherche et responsable du Laboratoire d'Informatique Systèmes et Télécommunications, de m'avoir accueilli au sein de son laboratoire, pour m'avoir encadré et soutenu dans cette thèse. Sa disponibilité, ses qualités humaines, ses conseils ont été indispensables à la concrétisation de cette recherche.

Je remercie également le président du jury Monsieur le Professeur Mohammed ADDOU ainsi que tous les membres de jury, les Professeurs : Fouad LAHJOMRI, Zine El Abidine ALAOUI ISMAILI, Oualid KAMACH, M'hamed AIT KBIR, Mohammed BOUHORMA pour l'honneur qu'ils m'ont fait en acceptant de rapporter et d'examiner ce travail.

Mes remerciements vont également à Monsieur le Professeur Abdeslam DRAOUI, Directeur du Centre des Etudes Doctorales pour m'avoir accueilli au CED, pour sa disponibilité et sa rigueur, et à Monsieur le Professeur Mohamed BOUHORMA, responsable du Laboratoire d'Informatique Systèmes et Télécommunications.

Mes remerciements vont également au Professeur Mohamed JBILOU, Chef du Département Génie Electrique, pour ses qualités humaines, son aide et son sérieux,

Je tiens à remercier chaleureusement mes collègues et amis du Département Génie Electrique et du laboratoire LIST pour la sympathie qu'ils ont témoigné à mon égard.

Enfin, merci à ma femme et à mes enfants de m'avoir supporté tout au long de la période de la thèse, sans leur soutien constant et leur patience, cette thèse ne serait probablement pas réalisée.

# Résumé

Le secteur industriel exige non seulement des performances des systèmes embarqués en termes de qualité, de productivité et de rentabilité, mais aussi en termes de sécurité. Ainsi, pour chaque fonction de sécurité, l'identification du niveau de sécurité requis (PL-*Performance level*) ou du niveau d'intégrité (SIL) représentant le niveau de crédibilité qui peut être accordé aux systèmes automatisés de sécurité (SAS) nécessite une analyse et une évaluation du risque. Dans ce sens, plusieurs méthodes ont été élaborées pour l'analyse du risque : méthode de l'arbre des causes, méthode de bloc-diagramme de fiabilité, méthode des chaînes de Markov. Ces méthodes sont toutes basées sur le calcul de la probabilité moyenne de défaillance sur demande PFDavg et permettent de qualifier les systèmes en question par un niveau de sécurité SIL (*safety integrity level*) exigé par les normes de sécurité telles que EN9100, norme européenne pour l'aéronautique et l'espace, et ISO/TS 16949, norme européenne pour l'automobile.

Les travaux présentés dans cette thèse s'inscrivent dans ce sens et s'intéressent plus précisément aux systèmes embarqués de contrôle-commande dédiés à la sécurité. Ils ont pour objectif de développer un prototype matériel-logiciel d'un contrôleur flou basé sur un moteur d'inférence floue (MIF) utilisant la technologie FPGA et répondant aux contraintes de sécurité avec une structure redondante au moins un parmi deux avec diagnostic (1oo2D) en temps réel sur une cible FPGA.

Nous faisons d'abord une évaluation quantitative de la fonction de sécurité pour le contrôleur flou à base de FPGA, en fonction des différentes architectures dédiées aux applications de la sécurité (1oo1, 1oo1D, 1oo2, 1oo2D, 2oo2) de la norme de sécurité CEI 61508 et à travers les différents modèles cités ci-dessus afin de caractériser ces architectures par un niveau de sécurité SIL. Les résultats obtenus montrent que le système d'inférence floue de structure 1oo1 est 49 fois plus confronté à une défaillance dangereuse qu'une architecture redondante 1oo2, et que le système d'inférence floue de structure 1oo2D est 100 fois moins confronté à une défaillance dangereuse qu'une architecture redondante 1oo2. La meilleure PFDavg et donc le meilleur SIL a été obtenue pour l'architecture « au moins un parmi deux avec diagnostic » (1oo2D). Ensuite, nous validons cette d'architecture (1oo2D) pour le MIF, par une analyse détaillée AMDEC et le calcul de la probabilité moyenne de défaillance sur demande PFDavg par les trois méthodes citées ci-dessus, et enfin son implémentation sur la cible FPGA.

**Mots clés** : Système d'inférence floue avec sécurité, Circuit FPGA, Langage VHDL, Les valeurs PFDavg et MTTF, L'architecture 1oo2D, Bloc-diagramme de fiabilité, Arbre des causes, Processus de Markov, Niveau d'intégrité (SIL).

# Abstract

The process industry requires not only the performance of embedded systems in terms of quality, productivity and profitability, but also in terms of safety. Thus, for each safety function, the identification of the security level requested (PL- Performance Level) or the integrity level (SIL) which representing the level of credibility that may be granted to a Safety Instrumented System (SIS), requires analysis and risk assessment. In this sense, several methods have been developed for the risk analysis: tree causes analysis, reliability block diagram and Markov analysis. These methods are all based on the calculation of the probability of failure on demand PFDavg and allow characterizing the systems by a Safety Integrity Level (SIL) required by safety standards such as EN9100 European standard for the Aeronautics and space and ISO/TS 16949 European standard for automobile.

The works presented in this thesis, are focuses specifically on design and implement a safety fuzzy logic controller. They aim to develop a hardware-software prototype of a fuzzy controller based on a fuzzy inference system using FPGA technology and meet the constraints of security with a redundant structure at least one two with diagnosis (1oo2D) in real-time on a FPGA target.

Firstly, a quantitative assessment of the safety function for the fuzzy inference system, depending on different architectures dedicated to security applications (1oo1, 1oo1D, 1oo2, 1oo2D, 2oo2) of the standard safety IEC 61508 and through the various models mentioned above to characterize these architectures by SIL.

The results show that the fuzzy inference system of 1oo1 structure is 49 times more confronted a dangerous failure that as a redundant architecture 1oo2 and the fuzzy inference system of 1oo2D structure is 100 times less confronted a dangerous failure that as a redundant architecture 1oo2. The best value of PFDavg and therefore the best security integrity level was obtained for the 1oo2D architecture. The validation of this architecture (1oo2D) for fuzzy inference system is done through a detailed analysis FMEA and calculating the average probability of failure on demand by three methods mentioned above and finally the system was implemented on the target FPGA with Xilinx software.

**Keywords:** Safety inference fuzzy system, FPGA, VHDL language, PFDavg and MTTF values, Redundant architecture 1oo2D, Reliability block diagram, Tree causes, Markov processes, Integrity Level SIL.

# Introduction générale

# Introduction générale

Les terminologies telles que la disponibilité, la sécurité et la sûreté de fonctionnement prêtent souvent à confusions par des utilisateurs non avisés, cependant le terme « disponibilité » indique la disposition d'un composant ou d'un système à être en état de marche à un instant donné, par contre le terme « sécurité » désigne l'aptitude d'une entité à ne pas conduire à des accidents intolérables. Plus précisément, la disponibilité est l'aptitude d'une entité à être en état d'exécuter une fonction sollicitée dans des conditions prédéfinies, à un instant défini, en supposant que la fourniture des moyens nécessaires est assurée ; alors que la sécurité, c'est l'aptitude d'un système à respecter, pendant toutes les phases de sa vie, un niveau acceptable de risques d'accident susceptible de causer une agression du personnel ou une dégradation majeure du système ou de son environnement [Wiki 13]. Selon les contextes, la disponibilité et la sécurité peuvent être des aptitudes compatibles ou antagonistes. Ainsi, si un produit ne dispose pas d'état de repli sûr en cas de panne (cas de l'avion en vol par exemple), la sécurité est obtenue par une forte disponibilité par une redondance. À l'inverse, si l'état de panne est plus sûr que l'état de fonctionnement (cas des transports terrestres, des systèmes ferroviaires par exemple), un haut niveau de sécurité peut entraîner une disponibilité médiocre, un compromis entre sécurité et disponibilité doit alors être trouvé.

Par contre, la sûreté de fonctionnement est l'aptitude d'une entité ou d'un système à satisfaire à une ou plusieurs fonctions requises dans des conditions données. Elle traduit la confiance qu'on peut accorder à un système. Au sens large, la sûreté de fonctionnement est considérée comme la science d'analyse de risque qui inclut les défaillances, les pannes, la disponibilité, la sécurité et la maintenance de système.

Les domaines industriels exigent non seulement des performances des systèmes en termes de qualité, de productivité et de rentabilité, mais aussi en termes de sécurité. La catastrophe de Seveso, en 1976, en Italie [WIK 00], qui a été déclenchée par la libération d'une quantité du gaz toxique — dioxine — dans l'air à cause d'une réaction chimique incontrôlée au niveau d'un réacteur de l'usine a eu lieu parce que le réacteur en question n'avait pas de système de refroidissement automatique ni de systèmes d'alerte.

Dans la gestion des entreprises, la sécurité industrielle, au sens large, consiste de façon générale à garantir la sécurité des biens, des personnes et également la pérennité de l'entreprise. Il s'agit alors de concilier les exigences de disponibilité, de rentabilité à court terme, avec les exigences de sécurité des biens et des personnes visant à réduire les risques, sur le plan environnemental, social et économique. Dans les entreprises industrielles, dont les activités présentent des dangers et donc des risques technologiques avérés ou plausibles, la sécurité industrielle se focalise alors sur l'analyse de ces risques et sur leur maîtrise.

Mais, en raison des interactions croissantes des groupes humains, avec les facteurs culturels associés, et des interconnexions techniques (systèmes de contrôle industriels, réseaux de télécommunications, développement technologique rapide), la sécurité industrielle est devenue de plus en plus complexe. Il était alors nécessaire de définir un référentiel comportant les règles générales de sécurité (normes, réglementations, lois, etc.) auxquelles les industriels ou les concepteurs des systèmes de sécurité doivent se conformer. Des réglementations sectorielles par type d'activité, souvent nationales, viennent compléter les référentiels généraux.

Par conséquent, diverses normes de sécurité sont apparues. Au cours des années quatre-vingt, l'Union européenne a adopté la directive Seveso I de « Système à grand risque ». Celle-ci a été remplacée par la directive Seveso II 96/827EU « Maîtrise des dangers liés aux accidents majeurs impliquant des substances dangereuses ».

En 1998, la norme de sécurité CEI 61508 [CEI 06] concernant la sécurité fonctionnelle des systèmes électriques, électroniques, et électroniques programmables qui couvre les fonctions de sécurité automatisées a été publiée. Ce sont des systèmes automatiques qui, en réponse à des signaux d'entrées, génèrent des ordres de mise en sécurité tels que par exemple des systèmes d'arrêt d'urgence, une détection de survitesse générant un freinage automatique, une détection de présence générant une mise en sécurité, un système de signalisation, de positionnement, dans des domaines très variés comme le médical, le nucléaire, le secteur des machines, les procédés continus, l'automobile, le ferroviaire, l'armement. Les systèmes automatisés de sécurité se caractérisent par l'utilisation de capteurs, d'automates, de réseaux de communication, des actionneurs, pour réaliser des fonctions de sécurité, qui devront, selon l'ampleur du risque couvert, garantir un certain niveau d'intégrité de sécurité (SIL) afin d'amener le risque à des limites de tolérances acceptables.

Pour chaque fonction de sécurité, l'identification du niveau de sécurité requis (PL — *performance level*) ou du niveau d'intégrité (SIL) représentant le niveau de crédibilité qui peut être accordé à ces systèmes automatisés de sécurité (SAS) nécessite une analyse et une évaluation du risque.

Ainsi, et depuis août 2002, la norme de sécurité CEI 61508 [CEI 06] a obtenu le label européen EN 61508. Cette nouvelle norme de sécurité définit pour la première fois les exigences de sécurité pour les systèmes automatisés de sécurité, quel que soit leur domaine d'application, en tenant compte pour la première fois aussi des systèmes à base de microprocesseurs.

La norme CEI 61508 a été adoptée dans le monde entier et a été mise dans le code du droit national dans de nombreux pays industriels (par exemple, en Australie on parle de la norme AS 61508, en Grande-Bretagne on parle de la norme BS CEI 61508, aux États-Unis on parle de la norme NFPA 79-2002, et au Japon on parle de la norme JIS C 0508).

Alors que la norme CEI 61508 s'applique à la fabrication et au développement de dispositifs de protection au niveau du matériel ainsi que le logiciel, la norme

CEI 61511, « Sécurité fonctionnelle des systèmes instrumentés de sécurité (SIS) pour l'industrie de procès » est dédiée aux opérateurs et aux planificateurs de dispositifs et équipements de sécurité. Elle fournit des recommandations et des lignes directrices pour évaluer le risque de dommages causés par les équipements et aide dans le choix des composants appropriés, liés à la sécurité.

La première étape dans la conception d'un système instrumenté de sécurité, c'est d'identifier quels sont tous les éventuels dangers auxquels seront exposés les opérateurs et l'environnement.

L'identification et la classification des risques permettent de définir le risque pour l'opérateur et l'environnement, c'est-à-dire la combinaison de la probabilité que le danger se produise et du type de dommages possibles.

Dans ce sens, plusieurs méthodes pour l'analyse du risque ont été élaborées. On distingue la méthode de l'arbre des causes [ISA 01], la méthode de bloc-diagramme de fiabilité [CEI 06] et la méthode des chaînes de Markov [GHY 08]. Ces méthodes sont toutes basées sur le calcul de la probabilité moyenne de défaillance sur demande PFDavg et permettent de qualifier les systèmes en question par un niveau de sécurité SIL (*safety integrity level*).

Dans cette thèse, une analyse de risque sera faite pour les contrôleurs flous, qui deviennent de plus en plus complexes. En effet, les systèmes d'inférence floue contiennent des technologies numériques exigeant des arbitrages matériel-logiciel. Dans le contrôleur flou, l'information d'un dispositif externe telle que celle contenue dans un capteur est convertie en un signal qui commande la sortie pour piloter un dispositif ou des dispositifs tels que des moteurs ou des actionneurs par le biais d'un moteur d'inférence floue [AM 13].

Les algorithmes de calcul utilisés dans les moteurs d'inférences des régulateurs flous sont généralement implémentés sur des calculateurs à base de microprocesseurs (calculateurs, PC, microcontrôleurs, circuits intégrés à application spécifique [ASIC]) et sur des plates-formes logicielles (Labview, Matlab, langage C, etc.) et dont l'architecture dépend de l'application. Ceci présente plusieurs difficultés :

- Ces algorithmes sont tous basés sur un ensemble de calculs mathématiques complexes et très coûteux en temps de calcul [KPY 98].
- Les concepteurs doivent concevoir des régulateurs flous avec de multiples domaines d'applications qui demandent un ensemble de compétences difficiles à réunir dans une même équipe de concepteurs.
- Les plates-formes de développement classiques, Labview, Matlab, Langage C, posent des problèmes de portabilité de ces applications.
- Les circuits intégrés à application spécifique (ASIC), dont les contrôleurs flous font partie, posent des problèmes de maintenance et de compatibilité à cause de leur délai de livraison par leur fabricant garanti pour vingt ans sur le marché.

Un changement de leur architecture après cette période nécessite un changement radical du système de régulation et de commande.

À ces difficultés de conception et de développement s'ajoutent les contraintes toujours plus fortes du marché actuel, qui exige la réduction du temps de conception et de la mise sur le marché.

Ces constatations exigent la mise en œuvre de nouvelles technologies de conception permettant de fiabiliser et de diminuer le cycle de conception et ceci, grâce à l'émergence de plates-formes, permettant le passage du prototypage virtuel au prototypage réel dans des délais plus courts, avec un coût plus faible. Ici, on parle de la technologie du circuit FPGA (*field programmable gate array*) qui continue de gagner du terrain sur le marché mondial. Le chiffre d'affaires mondial des FPGA est passé de 14 millions de dollars américains en 1993 à 2,75 milliards de dollars en 2010 [EIB 06].

Le succès des solutions basées sur des circuits intégrés utilisant les circuits FPGA — lesquelles, en peu de temps, ont constitué une réelle alternative au remplacement des circuits basés sur la technologie d'ASIC — est dû à :

- la rapidité de développement et au temps de mise sur le marché : *time-to-market* (TTM) ;
- la puissance de calculs numériques qui s'avère supérieure à celle des processeurs de signaux numériques (DSP) [PJA 90] ;
- la rapidité et à la fiabilité du passage de prototypage virtuel logiciel au prototypage matériel.

Cette puissance, cette flexibilité et cette disponibilité immédiate des circuits FPGA sont dues à la structure interne de la puce FPGA qui se compose de plusieurs petites unités de blocs logiques, également appelés blocs logiques programmables (CLB), lesquels, à leur tour, sont reliés entre eux par un réseau de connexions. Les unités possèdent également des blocs classiques de mémoire statiques (SRAM) pour stocker les données, ainsi que des entrées et des sorties. Tout cela a permis la réalisation de systèmes très complexes et performants.

Cependant, cette complexité matérielle provoque des défauts aléatoires qui doivent non seulement être détectés, mais aussi donner la possibilité de ramener le système à un état sûr en cas d'occurrence d'une défaillance aléatoire.

Les travaux présentés dans ce mémoire s'inscrivent dans la volonté de développer un prototype matériel d'un moteur d'inférence floue répondant aux contraintes de sécurité fonctionnelle avec le langage de description matériel VHDL (*hardware discription language*) sur une puce FPGA.

Ce manuscrit comporte une introduction générale et cinq chapitres principaux. La rédaction des chapitres suit un ordre logique permettant de mieux appréhender la problématique de la sécurité fonctionnelle dans les systèmes embarqués à base de la technologie FPGA. Ainsi, dans l'introduction générale, nous avons défini le contexte

scientifique et industriel dans lequel s'est déroulé ce travail et nous présentons les motivations de cette étude, les problèmes et les contraintes liées à la sûreté de fonctionnement des systèmes embarqués et en particulier les systèmes d'inférence floue.

Dans le premier chapitre, nous présentons un aperçu sur le fondement mathématique de la logique floue et nous décrivons également la structure d'un contrôleur à base d'un système inférence floue et sa mise en œuvre par le logiciels Xilinx .

Le chapitre 2 est consacré à la technologie FPGA, et à l'analyse détaillée de sa structure interne. Un intérêt particulier a été accordé à la présentation et à la modélisation des différentes erreurs, fautes et défaillances, qui peuvent se produire dans une cellule mémoire de genre statique (SRAM) ainsi que les différentes défaillances et les mécanismes de détection des fautes du matériel dans la puce FPGA et le langage de description matérielle VHDL. Cette chapitre nos permettra la compréhension de la nécessité de la structure redondante des systèmes basés sur la technologie FPGA.

Le chapitre 3 présente et définit les différentes terminologies nécessaires à la compréhension d'un système instrumenté de sécurité (SIS) : sécurité fonctionnelle, fiabilité, disponibilité, maintenabilité, sûreté de fonctionnement, taux de défaillance, niveau de performance requis, probabilité de défaillance dangereuse et le niveau de sécurité. Il présente également la structure des différentes architectures dédiées à ces systèmes à savoir les architectures un parmi un (*1 out of 1*), un parmi un avec diagnostic un. (*1 out of 1D*), un parmi deux (*1 out of 2*), un parmi deux avec diagnostic (*1 out of 2D*) et deux parmi deux (*2 out of 2*), ainsi qu'une analyse détaillée des pannes qui peuvent affecter le système d'inférence floue avec sécurité avec les tests de diagnostic et les tests d'inspection associés. Le chapitre se termine par la modélisation de calcul de la valeur de la probabilité de défaillance dangereuse d'un système avec différentes architectures en utilisant trois méthodes : modélisation par des blocs-diagrammes de fiabilité, modélisation à partir de l'arbre des causes, méthode qui consiste à modéliser le comportement du système en présence des fautes possibles et enfin, modélisation grâce à des graphes de Markov, qui décrivent les états intermédiaires du système.

Le chapitre 4 est consacré à l'évaluation qualitative du régulateur flou avec la représentation de l'architecture redondante homogène du système. Le chapitre fait l'analyse des différentes architectures dédiées aux applications de la sécurité (1oo1, 1oo1D, 1oo2, 1oo2D, 2oo2) du système d'inférence floue à partir de la détermination de la probabilité moyenne de défaillance sur demande PFDavg, du temps de l'indisponibilité et le facteur de réduction de risque pour chaque structure. Ceci afin de pouvoir choisir la structure qui répond le mieux aux exigences de la sécurité. Ainsi et en partant du graphe de risque, une estimation du risque et une allocation du niveau de performance pour maîtriser le risque ont été effectuées.

Dans le chapitre 5, nous faisons une analyse détaillée des modes de défaillances et effets au niveau des composants du système, le moteur d'inférence floue d'architecture (1oo2D) implémenté en FPGA : le convertisseur analogique-numérique ADC, le convertisseur numérique-analogique DAC et le circuit d'alimentation, ceci pour pouvoir faire la quantification du moteur d'inférence floue par le calcul de la probabilité moyenne de défaillance sur demande PFDavg par la méthode de l'arbre des causes, par la méthode du bloc-diagramme de fiabilité et par la méthode des chaînes de Markov. La mise en œuvre du moteur d'inférence floue sur l'architecture 1oo2D sur la cible FPGA sera aussi présentée. À la fin on présentera nos conclusions et perspectives.

# Chapitre 1

# Moteur d’inférence floue

## *Résumé du chapitre 1 :*

*Dans ce chapitre, nous présenterons un aperçu sur le fondement mathématique de la logique floue et nous présenterons également la structure du système flou et sa mise en œuvre par le logiciel de Xilinx*

# 1 Moteur d'inférence floue

## 1.1 Introduction

De nos jours, la logique floue est un axe de recherche important sur lequel se focalisent de nombreux scientifiques. Des retombées technologiques sont d'ores et déjà disponibles, tant dans le domaine grand public (machines à laver, appareils photo, etc.) que dans le domaine industriel (réglage et commande de processus complexes liés aux automobiles, et à l'énergie) [KPY 98].

La logique floue présente une logique graduelle qui fait adapter à la logique numérique une formalisation naturelle linguistique très proche de notre perception nuancée du monde représentant l'interface qualitative linguistique et le numérique quantitatif. Le langage naturel abonde d'expressions « floues » comme : « la température est fraîche » ou bien « élevée », « la vitesse est excessive », « la distance est petite », etc. Le traitement de ce genre de données contenant de l'imprécision, de l'incertitude ou de la subjectivité exige des concepts, des techniques et des méthodes formellement rigoureux pour acquérir et traiter ces données floues.

La difficulté d'interprétation de ces variables linguistiques implique nécessairement une mise en contexte. La modélisation des concepts est réalisée par la théorie de sous-ensembles flous introduite par M. Lotfi Askar-Zadeh [LAZ 00], professeur à l'Université de Californie à Berkeley, qui a développé une approche basée sur la théorie des sous-ensembles flous (*fuzzy set*), généralisant la théorie des ensembles classiques.
Celles-ci ont été le point de départ de beaucoup de travaux théoriques et pratiques dans le domaine des mathématiques floues, de traitement de l'incertitude des systèmes, de la représentation des connaissances de la modélisation et de la commande, etc.

## 1.2 Logique classique et la logique floue

Tous les microprocesseurs d'aujourd'hui reposent sur la logique booléenne. C'est-à-dire qu'ils ne prennent que deux valeurs : 0 ou 1, mais dans la langue naturelle, il y a beaucoup de termes qui renvoient à l'imprécis et à l'incertain tel que « vague » ou « général ». Ces termes sont une forme d'imprécision au langage dans la mesure où elles renvoient à plusieurs contextes ou référentiels possibles [AMB 13]. La logique floue propose de remplacer les variables booléennes par les variables floues.

## 1.3 Champ d'application de la logique floue

Les applications de la logique floue et les plates-formes matérielles utilisées pour la logique floue sont diverses, allant des microcontrôleurs à l'automate via des systèmes de contrôle par ordinateur et de procédés dans différents domaines de l'industrie. Le Tableau 1-1 : représente une partie du domaine d'application de la logique floue.

| Domaine | Description de l'application de la logique floue |
|---|---|
| Automobile | Contrôle de l'airbag, de la boîte de vitesse automatique, de la climatisation. |
| Entreprise | L'évaluation du personnel dans une grande entreprise. |
| Industrie chimique | Le contrôle du pH, de séchage et les processus de distillation chimique. |
| Militaire | La reconnaissance de la cible sous-marine, la reconnaissance automatique des cibles des images infrarouges thermiques. |
| Électronique | Contrôle de l'exposition automatique de caméra vidéo, les systèmes de climatisation. |
| Finance | Contrôle de transferts de billets de banque, gestion de fonds, prévisions boursières. |
| Industriel | Le contrôle des fours des cimenteries, le contrôle de l'échangeur de chaleur, le contrôle de l'usine de purification de l'eau, le contrôle des problèmes de satisfaction des contraintes dans la conception structurelle, le contrôle des usines de purification de l'eau. |
| Médical | Système d'aide au diagnostic médical. |

Tableau 1-1 : Domaine d'application de la logique floue

## 1.4 Sous-ensembles flous

Dans la théorie booléenne (logique binaire), l'appartenance d'un élément à un sous-ensemble est définie par une valeur logique précise, soit 1 si l'élément appartient au sous-ensemble ou bien 0 si non (Figure 1-1- a). Dans la logique floue, un élément peut appartenir à un sous-ensemble [AMB 14] avec un degré d'appartenance qui est décrit par une valeur comprise entre 0 et 1.

La transition entre les états (0) et (1) est maintenant progressive, on dit que la logique floue adoucit la logique binaire (Figure 1-1-b). La fonction représentée doit être considérée comme un degré d'appartenance. Ainsi, une chambre présente une température de 20 $^{o}$C, n'a qu'un faible degré d'appartenance de 17 % à l'état « froid » alors que celui-ci présente un fort degré d'appartenance de 75 % à l'état chaud.

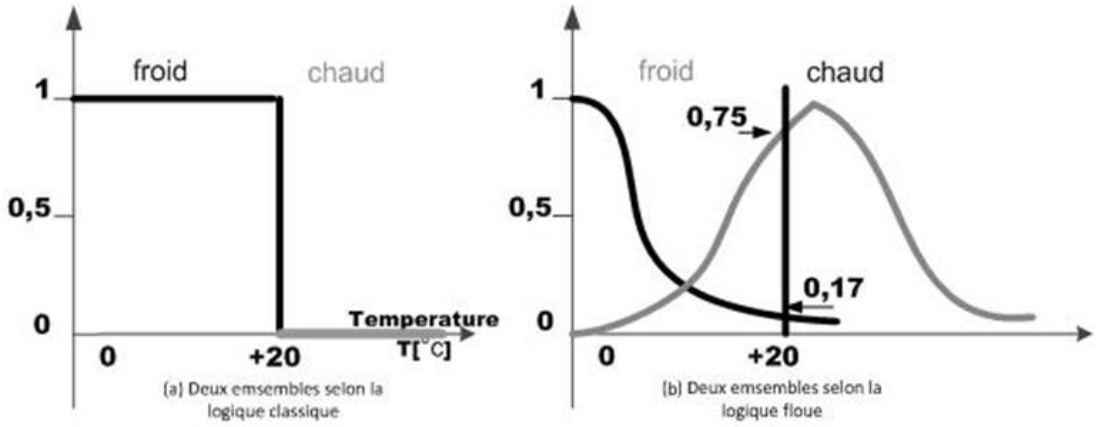



Figure 1-1 : Représentation graphique des sous-ensembles

### 1.4.1 Définition

Un sous-ensemble flou A dans un univers du discours X est caractérisé par sa fonction d'appartenance $\mu_A(x)$ de l'ensemble flou A qui associe à chaque élément x de l'univers de discours X une valeur dans l'intervalle [0,1].

$\mu_A(x)$ : X $\rightarrow$ [0,1] est définie comme suite [AMB 13] :

$$\forall x \in X,\ \mu_A(x) \in [0,1], \qquad \mu_A(x) = \deg ré(x \in A) \tag{1-1}$$

### 1.4.2 Opérations de base sur les sous-ensembles flous

Supposons que A et B sont deux sous-ensembles flous définis dans un univers du discours X par les fonctions d'appartenance $\mu_A(x)$ et $\mu_B(x)$. On peut définir des opérations ensemblistes telles que l'égalité, l'inclusion, l'intersection, l'union et le complément grâce à des opérations sur les fonctions d'appartenances.

**Égalité** : A et B sont dits égaux, propriété que l'on note A = B, si leurs fonctions d'appartenance prennent la même valeur en tout point de X :

$$\forall x \in X \quad \mu_A(x) = \mu_B(x). \tag{1-2}$$

**Inclusion** : A est dit inclus dans B, propriété que l'on note A $\subseteq$ B, si tout élément x de X qui appartient à A appartient aussi à B avec un degré au moins aussi grand [AMB 13] et [LAZ 65] :

$$\forall x \in X \quad \mu_A(x) \leq \mu_B(x). \tag{1-3}$$

Les définitions d'intersection, d'union et de complément de sous-ensembles flous définies ci-dessous font intervenir les opérateurs de minimum, maximum et de complémentation à 1.

**Intersection** : l'intersection de A et B, que l'on note A∩B, est le sous-ensemble flou constitué des éléments de X affectés du plus petit des deux degrés d'appartenance $\mu_A(x)$ et $\mu_B(x)$ [AMB 13] et [LAZ 65] :

$$\forall x \in X \quad \mu_{A\cap B} = \min(\mu_A(x), \mu_B(x)). \tag{1-4}$$

**Union** : l'union de A et B, que l'on note A∪B, est le sous-ensemble flou constitué des éléments de X affectés du plus grand des deux degrés d'appartenance $\mu_A(x)$ et $\mu_B(x)$ [AMB 13] et [LAZ 65] :

$$\forall x \in X \quad \mu_{A\cup B} = \max(\mu_A(x), \mu_B(x)). \tag{1-5}$$

**Complément** : le complément de A, que l'on note $A^c$, est le sous-ensemble flou de X constitué des éléments x lui appartenant d'autant plus qu'ils appartiennent peu à A [AMB 13] et [LAZ 65] :

$$\forall x \in X \quad \mu A^C = 1 - \mu_A(x). \tag{1-6}$$

D'autres définitions sont également possibles lorsque l'on fait intervenir les concepts de normes triangulaires (ou t-normes) et de conormes triangulaires (ou t-conormes) [AMB 13] et constituent une généralisation des opérations de combinaison de type minimum ou maximum. Le Tableau 1-2 représente les t-normes et t-conormes les plus utilisées [AMB 14] :

| **Dénomination** | **t-norme** | **t-conorme** |
|---|---|---|
| Zadeh | min(x,y) | max(x,y) |
| Probabiliste | x.y | X+y-x.y |
| Lukasiewie | max(x+y-1,0) | min(x+y, 1) |
| Hamacher $\gamma \geq 0$ | $\frac{xy}{\gamma+(1-\gamma)(x+y-xy)}$ | $\frac{x+y-xy-(1-\gamma)xy}{1-(1-\gamma)xy}$ |
| Weber | $\begin{cases} x & si \quad y=1 \\ y & si \quad x=1 \\ 0 & \sin on \end{cases}$ | $\begin{cases} x & si \quad y=0 \\ y & si \quad x=0 \\ 0 & \sin on \end{cases}$ |

Tableau 1-2 : Dénomination t-normes et t-conormes

## 1.5 Raisonnement en logique floue

Dans la logique classique, les méthodes de déduction sont certaines et formalisées. Dans le cadre de la logique floue, il est possible de généraliser les méthodes de raisonnement lorsqu'on dispose de connaissances incertaines ou imprécises.

### 1.5.1 Variables linguistiques

La logique floue repose sur le concept des variables linguistiques et les fonctions d'appartenances [AMB 13]. En effet, la description d'une certaine situation, ou un phénomène par les variables linguistiques permettent de décrire un cadre très général de la connaissance acquise sur une variable. Une variable linguistique est caractérisée par un quintuple (x, T(x), U, G, M), dans lequel [AMB 13] :

- x est le nom de la variable définie sur l'univers du discours X,
- T(x) = A1, A2…, An est l'ensemble des valeurs linguistiques que peut prendre x,
- U est l'univers du discours associé avec la valeur de base,
- G est la règle syntaxique pour générer les valeurs linguistiques de x,
- M est la règle sémantique pour associer un sens à chaque valeur linguistique.

### 1.5.2 Implications floues

Dans la proposition floue « Si (X est A) ; alors (Y est B) », les propositions (X est A) et (Y est B) , sont construites à partir des deux variables linguistiques (x, T(x), U, G, M) et ((y, T(y), U, G, M),) qui sont *a priori* indépendantes. L'implication floue permet de définir une liaison entre la prémisse Si (X est A) et la conclusion (Y est B) de cette règle. Les implications floues le plus souvent employées sont précisées dans le Tableau 1-3 [MER 08]. La définition des implications floues peut théoriquement faire intervenir n'importe quel expert, pourtant, parmi elles on utilise souvent les implications de Mamdani [MAM 77] et de Larsen [BBM 07] qui dominent les applications industrielles dans la commande floue.

| **Dénomination** | **Valeur de vérité** | $I(\mu_A(x), \mu_B(x))$ |
|---|---|---|
| Mamdani | $I_m$ | $\min(\mu_A(x), \mu_B(x))$ |
| Larsen | $I_l$ | $\mu_A(x) \times \mu_B(x)$ |
| Reichenbach | $I_r$ | $1 - \mu_A(x) + \mu_A(x) \times \mu_B(x)$ |
| Willamette | $I_w$ | $\max(1 - \mu_A(x), \min(\mu_A(x), \mu_B(x))$ |
| Rescher-Gaines | $I_{rg}$ | $\begin{cases} 1 & si \quad \mu_A(x) \le \mu_B(x) \\ 0 & \end{cases}$ |
| Brouwer-Godel | $I_{bg}$ | $\begin{cases} 1 & si \quad \mu_A(x) \le \mu_B(x) \\ \mu_B(x) & \end{cases}$ |
| Luksiewiez | $I_l$ | $\min(1 - \mu_A(x) + \mu_B(x), 1)$ |

Tableau 1-3 : Les implications floues les plus utilisées

### 1.5.3 Contrôleur flou

D'une façon générale, un système de commande flou a pour objectif de piloter un processus afin d'obtenir un fonctionnement correct de ce dernier (régulation d'une grandeur physique par exemple). Si on dispose d'un modèle plus ou moins précis du système à commander, on peut utiliser un contrôleur ou bien un régulateur classique (PID par exemple), mais lorsque le système est difficilement modélisable ou bien lorsqu'il s'agit d'un système non linéaire, la conception du système de régulation avec un contrôleur classique reste très difficile.

Lorsqu'un opérateur humain commande manuellement un système, les actions qu'il réalise sont dictées par une connaissance subjective du fonctionnement de ce système. Par exemple, si l'eau est chaude dans une piscine, on la refroidit, si elle est très chaude on la refroidit plus. Cette commande du système peut être envisagée de façon différente selon la personne qui la réalise : la sensation de la chaleur n'est pas directement liée à une mesure de la température.

Si on veut décrire la température de l'eau de la piscine par une variable linguistique, on peut utiliser l'ensemble des termes suivant : T(x)= {froid, frais, chaud}. En considérant que l'univers du discours de la variable température est l'intervalle [0, 50 $^{o}$C], on peut utiliser les règles sémantiques suivantes pour définir les termes linguistiques :

- « froid » est « une température inférieure à environ 15 $^{o}$C » ;
- « frais » est « une température d'environ 20 $^{o}$C » ;
- « chaud » est « une température d'environ 25 $^{o}$C ».

Ces termes peuvent être caractérisés par les fonctions d'appartenance représentées sur la Figure 1-2.

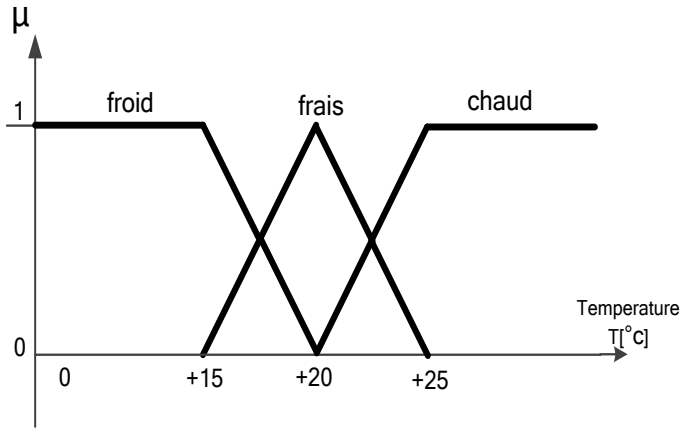



Figure 1-2 : Variable linguistique pour décrire la température

La mesure réalisée sur le système est prise en compte par l'intermédiaire d'une variable qui prend des valeurs linguistiques {froid, frais, chaud}, qui est issue d'une analyse faite par un expert. Ensuite, l'action à réaliser est déduite à la fois d'un ensemble de règles de commande de type « si… alors » (s'il fait froid, alors, on chauffe plus) liant des variables floues d'entrée à une variable floue de sortie. Le traitement conduit à un résultat flou qu'il faudra traiter pour obtenir l'information utile, capable d'attaquer l'interface de commande [AMB 13].

Le moteur d'inférence floue est formé de trois étapes essentielles comme indiqué sur la Figure 1-3.

La première phase de fuzzification transforme les valeurs numériques en degrés d'appartenance aux différents ensembles flous de la partition. La seconde phase

concerne le module de raisonnement flou qui est constitué de deux blocs, soit le moteur d'inférence et les bases de règles floues. Enfin, une phase de défuzzification permet d'inférer une valeur précise, utilisable par une commande.

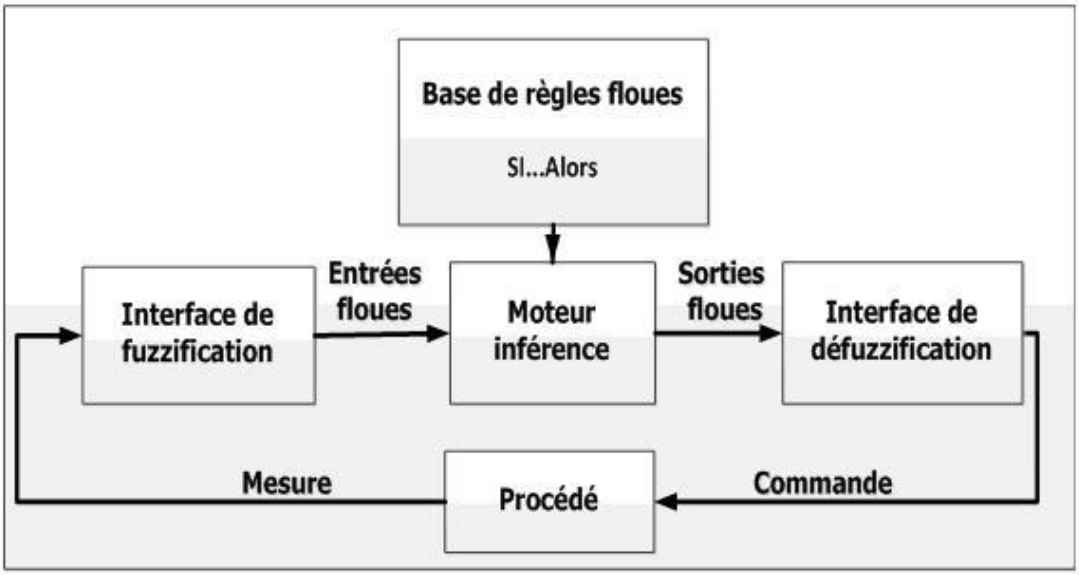



Figure 1-3 : Structure générale d'un moteur d'inférence floue

### 1.5.3.1 Fuzzification

Le premier traitement des données soumises dans l'interface d'entrée du contrôleur flou est l'opération de fuzzification, soit la transformation de données non floues en données floues par le calcul du degré d'appartenance de la valeur mesurée pour chaque fonction d'appartenance. Durant cette phase de normalisation, chaque mesure issue du système est modifiée pour fournir une valeur appartenant à un univers du discours. Il s'agit donc d'attribuer à chaque variable des degrés d'appartenance aux différents états que l'on doit définir. La normalisation est souvent réalisée par transformation linéaire. Enfin, les valeurs normalisées déduites de chacune des entrées sont transformées en qualifications linguistiques.

### 1.5.3.2 Règles floues

Les règles floues permettent de déduire des connaissances concernant l'état du système en fonction des qualifications linguistiques fournies par l'étape de fuzzification. Ces connaissances sont traduites en règles simples pouvant être utilisées dans un processus d'inférence floue.

### 1.5.3.3 Inférence floue

L'inférence floue est une relation floue définie entre deux sous-ensembles. Les inférences floues définies par Mamdani et Surgeno [BBM 07] sont les plus souvent utilisées.

L'inférence floue proposée par Mamdani stipule que lorsque les conditions sont liées par une logique « ou », on considère le degré d'appartenance maximum parmi les conditions d'entrée, par contre si les conditions sont liées par une logique « et », on considère le degré d'appartenance minimum parmi les conditions. L'inférence max-min est une méthode relativement simple largement utilisée dans les applications industrielles.

L'inférence floue proposée par Takagi-Sugeno garantit une continuité de la sortie. Cette méthode d'inférence s'avère très efficace dans des applications faisant intervenir à la fois des techniques linéaires, d'optimisation et adaptatives.

#### 1.5.3.4 Défuzzification

La méthode d'inférence vue dans la section précédente fournit un résultat basant sur une fonction d'appartenance, or, la sortie du contrôleur est une grandeur continue. La phase de la défuzzification consiste en la transformation des résultats flous en sorties précises en définissant une correspondance entre le résultat de l'inférence floue et la grandeur continue en sortie.

Plusieurs techniques de défuzzification ont été élaborées, la méthode du centre de gravité est la plus utilisée et consiste à tracer, sur un même diagramme, des différentes zones trapézoïdales correspondant à chacune des règles, et à calculer le centre de gravité de la zone consolidée.

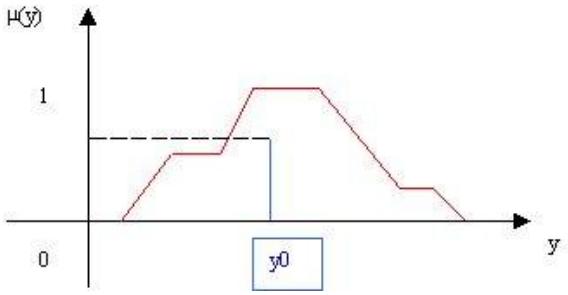



Figure 1-4 : La méthode du centre de gravité

Dans le cas où les sous-ensembles flous F, relatifs aux commandes à appliquer dans les règles floues, sont des singletons, la méthode de défuzzification barycentrique se ramène à la transformation :

$$y_{cg} = \frac{\sum_{i=1}^{m} \mu_{R_i} \times y_i}{\sum_{i=1}^{m} \mu_{R_i}} \quad (1\text{-}7).$$

Avec

$\mu_{R_i}$ : degré de vérité de la variable floue de « sortie » pour chaque règle d'indice i

$y_i$ : valeur de la variable floue de « sortie » d'indice i

Cette méthode est de loin la plus coûteuse en puissance de calcul, et donne les meilleurs résultats. La puissance des processeurs disponibles aujourd'hui, dont certains dédiés à la logique floue, ne pose plus de problème pour l'implémentation de cette technique.

## 1.6 Technologie des systèmes d'inférences floues

### 1.6.1 Moteur d'inférence floue à base de microprocesseur

La société Inform GmbH au Portugal en étroite collaboration avec la société Texas Instruments, ont pu développer le microcontrôleur à logique floue du genre DSP le fuzzy TECH MCU-320 Édition [FUZ 00] mis en œuvre avec les solutions floues adaptées à des applications dédiées aux systèmes embarqués. Le Tableau 1-4 représente le type de contrôleurs flous commercialisés par la société Texas Instrument (TI) avec leurs caractéristiques.

| Microcontrôleur | Nombre de règles | Nombre d'entrées | Nombre de sorties | Langage de programmation |
|---|---|---|---|---|
| Fuzzytech professionnel édition, 16 bits résolution, Fréquence 33 MHz | 7 | 2 | 3 | Langage C |

Tableau 1-4 : Caractéristique du contrôleur flou « Fuzzytech » [FUZ 00]

### 1.6.2 Moteur d'inférence floue à base du circuit FPGA

L'application des circuits FPGA pour les systèmes d'inférence floue se révèle très bénéfique. En effet, les caractéristiques principales d'un moteur d'inférence floue peuvent être optimisées sans pour autant se soucier de l'influence d'un autre paramètre. En ce qui concerne la partie matérielle, la modification du circuit n'implique pas des dépenses dans un matériel supplémentaire, car il y aura seulement une

exploitation des ressources internes au FPGA. En particulier la possibilité de mise à jour ou de correction du circuit sans pour autant que ce soit compromettant pour la chaîne de fabrication ni pour le coût de revient du moteur d'inférence floue. L'approche proposée par [SKU 12] consiste à implémenter un moteur d'inférence floue en FPGA pour contrôler la vitesse d'un moteur, car les caractéristiques non linéaires telles que le frottement et la saturation d'un moteur, dégradent les performances des contrôleurs traditionnels [SKU 12]. Une autre application proposée par [DPY 11] concerne la mise en œuvre d'un moteur d'inférence floue pour des systèmes cellulaires mobiles afin de maintenir une communication fiable.

Sachant que les modèles de faute sont des fautes aléatoires du matériel dans un circuit FPGA et que chaque type de modèle de faute peuvent correspondre à plusieurs origines physiques différentes, ceci nous laisse conclure que ces applications ne sont pas fiables, car dans toutes ces applications les modèles de faute dans un circuit FPGA ne sont pas prises en considération. Une erreur dans un élément de la puce FPGA rend le système dans un état dégradé, et voire un état grave.

## 1.7 Conclusion

Nous avons présenté un aperçu sur le fondement mathématique de la logique floue et la technologie utilisée pour la mise en œuvre matérielle du moteur d'inférence floue. Le moteur d'inférence floue est formé de trois étapes, la phase de fuzzification et la phase de raisonnement flou qui est constitué de deux blocs, soit le moteur d'inférence et les bases de règles floues. Enfin, une phase de défuzzification permet d'inférer une valeur précise, utilisable par une commande. La réalisation matérielle et logicielle des moteurs d'inférence floue s'effectue dans le secteur industriel sur des microprocesseurs, des circuits intégrés spécifiques à une application (ASIC) qu'ainsi que sur des circuits de réseau de portes programmables (FPGA).

# Chapitre 2

# Réseau de portes programmables FPGA et leur modèle de fautes

## *Résumé du chapitre 2 :*

*Le chapitre 2 est consacré à la technologie FPGA, et à l'analyse détaillée de sa structure interne. Un intérêt particulier a été accordé à la présentation et à la modélisation des différentes erreurs, fautes et défaillances, qui peuvent se produire dans une cellule mémoire de genre statique (SRAM) ainsi que les différentes défaillances et les mécanismes de détection des fautes du matériel dans les circuits FPGA.*

# 2 Réseau de portes programmables FPGA

## 2.1 Introduction

Un circuit FPGA est un composant électronique qui contient des milliers, voire des millions de transistors connectés ensemble pour réaliser des fonctions logiques simples telles que des additions ou des soustractions ou des fonctions complexes telles que la mise en œuvre d'un contrôleur DSP (*digital signal processor*) ou bien un triac-microprocesseur. Ces circuits FPGA sont largement utilisés dans les systèmes embarqués notamment dans les domaines de l'automatisme, de l'automobile, de l'aéronautique et des télécommunications.

L'évolution des circuits logiques programmables a commencé par la création des réseaux logiques programmables (PAL : *programmable array logic*) qui présentaient l'avantage de réduire l'encombrement et la création de fonctions logiques personnalisées, ensuite vient l'étape des circuits programmables et effaçables (EPLD : *erasable programmable logic device*) qui présentent l'avantage de l'écriture électrique. Puis les circuits FPGA qui représentent une technologie qui permet de reprogrammer le circuit à la carte (situ). En effet l'avantage majeur que présentent les circuits FPGA, est leur grande flexibilité, car la structure interne peut être changée sans avoir à modifier la structure globale du circuit. Cet avantage nous permet de faire des prototypages rapides et de moindre coût en comparaison avec les circuits ASIC pour lesquels il faut des mois pour réaliser un prototype sans avoir la certitude qu'il puisse être opérationnel, en plus de cela, la moindre erreur nécessite de refaire le travail depuis le début.

Xilinx, Altera et Quicklogic sont les pionniers dans les domaines des circuits FPGA. Toutes ces compagnies se partagent le même concept architectural [TAV 96]. Il se divise en trois parties (Figure 2-1) :

- les blocs d'entrées et sorties (IOB) ;
- les blocs logiques de configuration (CLB) ;
- les interconnexions.

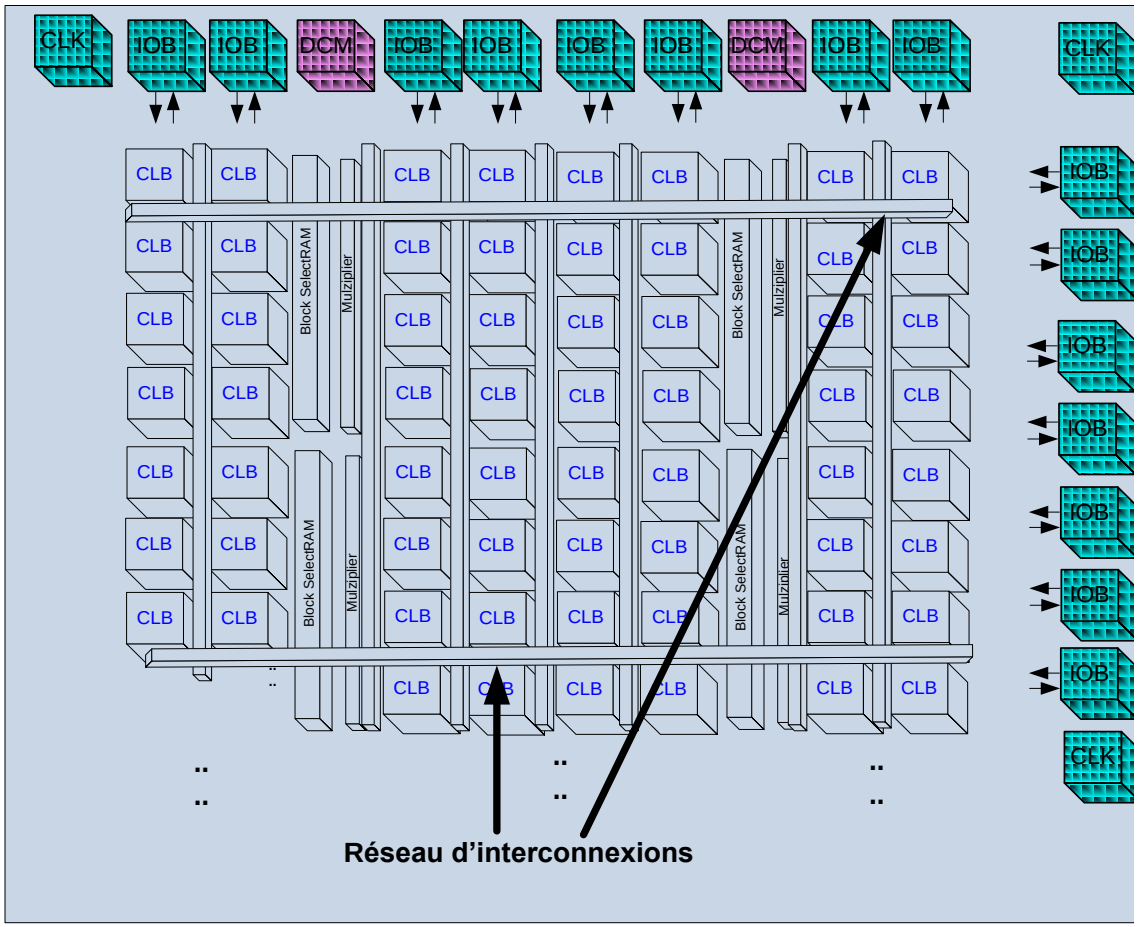



Figure 2-1 : Concept architectural de base des FPGA [XIL 12]

## 2.2 FPGA reconfigurables et non reconfigurables

Les circuits FPGA sont des éléments logiques programmables. À l'aide de blocs logiques préconstruits et de ressources de routages programmables, on peut mettre en œuvre des fonctionnalités matérielles personnalisées, sans avoir jamais besoin d'utiliser une maquette ou un fer à souder. Il suffit de développer des tâches de traitement numérique par le logiciel et les compiler sous forme de fichier de configuration ou de flux de bits (*bit stream*) contenant des informations sur la manière dont les composants doivent être reliés. En outre, la plupart des circuits FPGA sont totalement reprogrammables et peuvent adopter instantanément une nouvelle application digitale si une nouvelle configuration du circuit est recompilée.

À l'intérieur, le FPGA contient plusieurs petites unités de blocs logiques de base, également appelées CLB (*configurable logic blocks*), lesquelles, à leur tour, sont reliées entre elles par un réseau de connexions qui permet ainsi un routage universel.

Les circuits FPGA contiennent également des blocs de mémoire statique, SRAM, pour stocker les données.

Certains FPGA contiennent des PLL/DLL (*phase locked loop, delay locked loop*) pour fournir les signaux d'horloge, des traitements d´impulsions (DCM : *digital clock manager*) de même que de simples ALU (*arithmetic logic unit*), ce qui permet l'implémentation et la mise en œuvre de logiques plus complexes.

Il existe deux genres de circuit FPGA. Les FPGA non reconfigurables qui utilisent la mémoire anti-fusible et exigent très peu d'espace sur la puce, ils ont une résistance très faible au rayonnement naturel de même qu'une faible capacité. La programmation de ce type de FPGA prend du temps et ne peut être effectuée qu'une seule fois. Le programme existe toujours lors d'un redémarrage.

Par contre les FPGA à base de mémoire statique sont reprogrammables et utilisent les mémoires statiques SRAM dans la couche de configuration et la couche opérative, ce qui permet ainsi une description rapide et fréquente. L'inconvénient de la technologie de mémoire reconfigurable est la grande empreinte de la mémoire SRAM qui est nécessaire sur la puce. Après chaque redémarrage, les FPGA reconfigurables doivent être réécrits.

### 2.2.1 Architecture FPGA à base de mémoire statique SRAM

La technologie des FPGA à base de mémoire statique permet de configurer les interconnexions et de programmer les cellules logiques. Elles ont l'avantage d'être assez compactes et très rapides, mais leur inconvénient est qu'elles sont sensibles aux radiations et aux rayonnements cosmiques (Soft-Erros). La couche opérative (Figure 2-2) permet au FPGA de réaliser des applications à l'aide des composants combinatoires tandis que la couche de configuration rassemble les points mémoires SRAM chargés d'activer ou non les composants applicatifs. Une description détaillée des différentes couches est représentée dans ce mémoire [DSF 11].

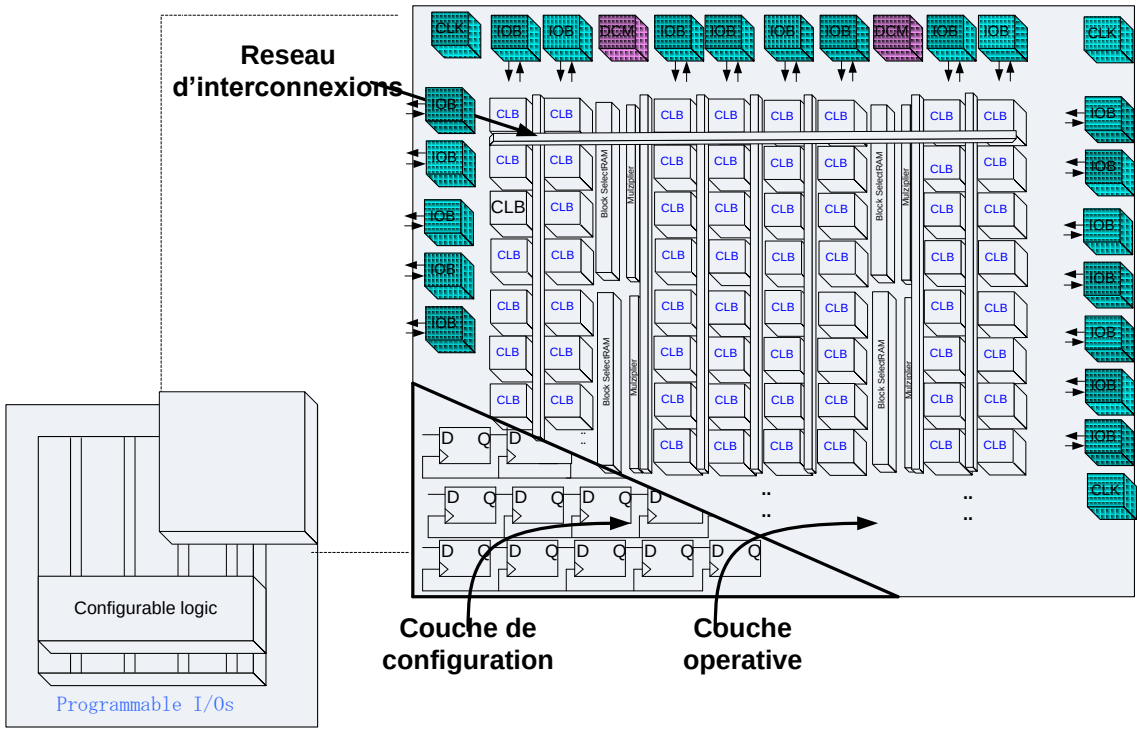



Figure 2-2 : Architecture FPGA à base SRAM

Le circuit FPGA à base de SRAM peut contenir des millions de bits SRAM, ce qui rend cette architecture plus performante et plus puissante. Le contenu de la cellule mémoire SRAM peut contenir des erreurs aléatoires susceptibles de mener le système implémenté dans la puce FPGA à des défaillances dangereuses. La question qui se pose donc est de savoir quel est le modèle de défauts pour cette nouvelle technologie FPGA à base SRAM. Quels sont les moyens de test qui peuvent être adaptés pour leur détection et pour leur diagnostic ?

### 2.2.2 Couche de configuration

En général, les circuits FPGA à base de mémoire statique ont tous une architecture similaire et peuvent saisir des millions de cellules SRAM de configuration. À la mise sous tension, chaque cellule SRAM dans la puce FPGA reçoit un seul bit de configuration. L'architecture des cellules mémoires dans les FPGA à base de mémoire SRAM est représentée sur la Figure 2-3 :

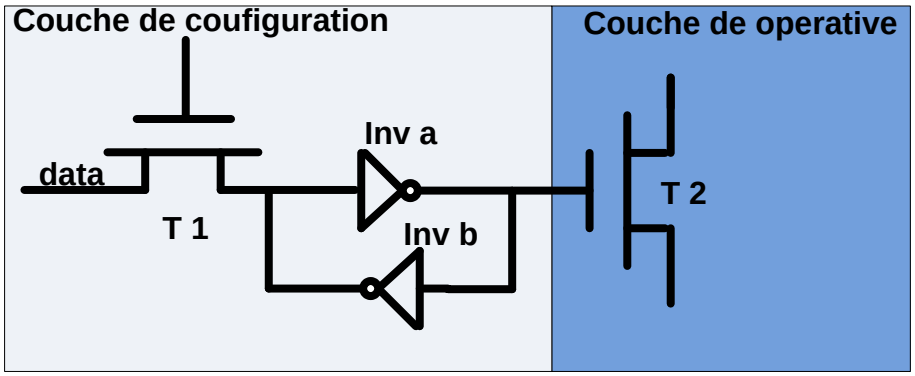



Figure 2-3 : La couche de configuration

Une cellule SRAM de configuration est constituée de deux inverseurs rebouclés (Inv a et Inv b). Selon [XIL 52], l'information est écrite (phase de configuration) ou lue (la fonctionnalité Readback) à partir de transistor d'accès T1 qui commande le transistor T2 appartenant à la couche opérative. La configuration des FPGA à base SRAM peut être réalisée à partir d'un mode Master, FPGA lit les données de configuration à partir d'un composant externe (une puce mémoire flash) qui mémorise un fichier du format BIT ou bien à partir d'un mode Slave, FPGA est configuré à partir d'un dispositif Master externe tel qu'un processeur. Cela peut généralement être effectué par l'intermédiaire d'une interface de configuration ou bien via l'interface JTAG Boundary scan.

### 2.2.3 Adressage des cellules SRAM de configuration

L'adressage des cellules SRAM de configuration consiste à connaître l'architecture et son placement dans la puce FPGA. Les cellules SRAM sont réparties régulièrement parmi les éléments de la couche opérative. La Figure 2-4 représente le principe de configuration et d'adressage de ces dernières.

L'adressage d'une cellule SRAM se fait à partir de la ligne de données et d'adressage. Cependant, une cellule d'un même rang partage la même ligne d'adressage et une cellule d'une même colonne partage la même ligne de données.

Après que le registre de décalage des données (RDD) de longueur n soit chargé par les bits de configuration, ils seront donc écrits en parallèle dans les cellules d'une même ligne, au travers des n ligne de données. L'écriture d'une ligne est vérifiée ou bien est validée par le registre au décalage des adresses (RDA) de longueur m. Selon [REA 00] la fonction Readback nous permet de vérifier le contenu de la couche de configuration. Elle fonctionne selon le même principe que l'écriture. La Figure 2-5 représente l'architecture d'adressage des données à l'aide des registres RDD et RDA [DSF 11].

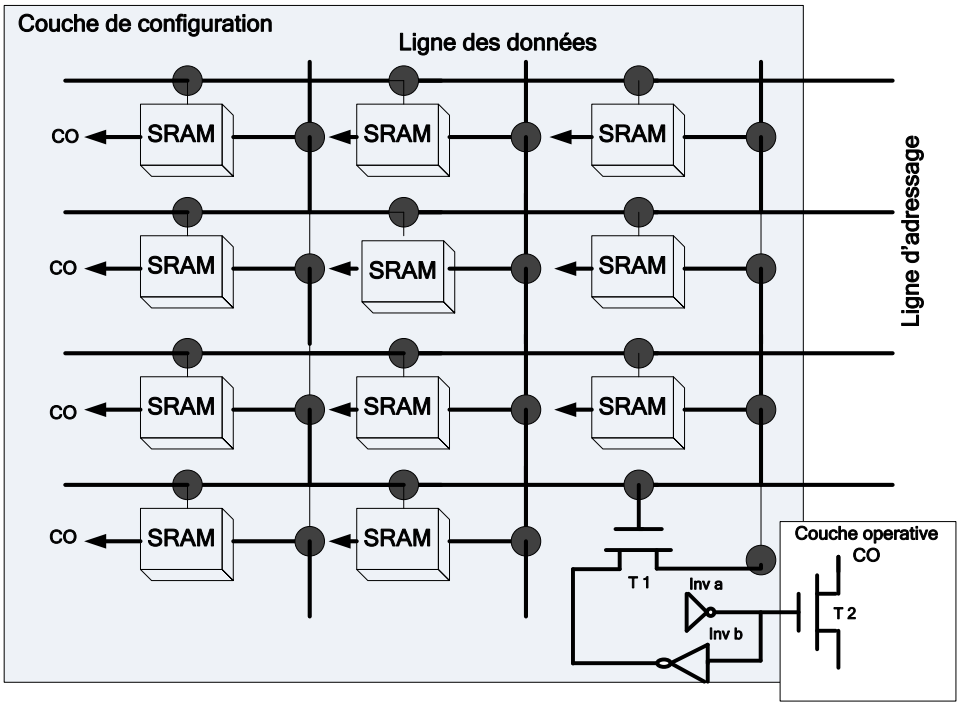



Figure 2-4 : Matrice de configuration SRAM

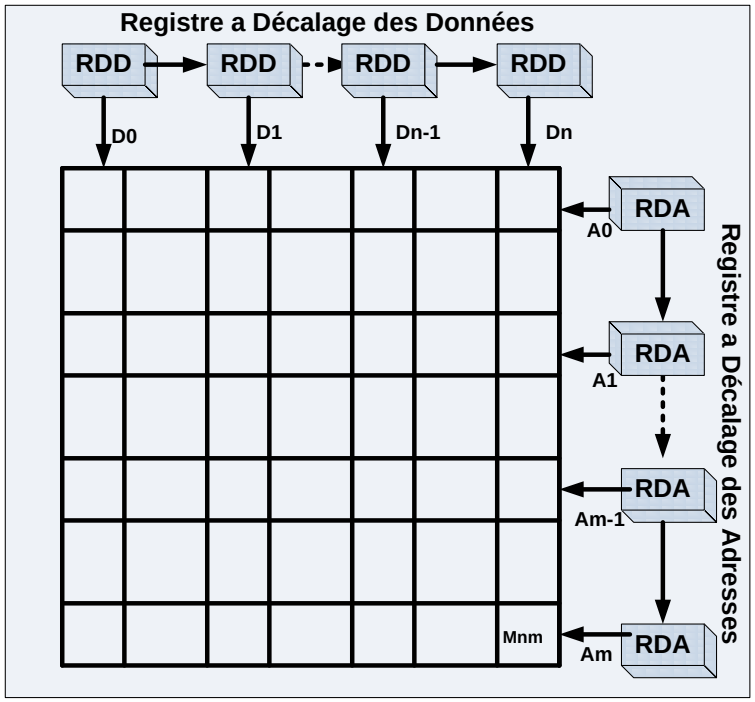



Figure 2-5 : L'adressage des données dans la couche de configuration

### 2.2.4 Couche opérative

La couche opérative de FPGA à base SRAM dispose d'une interface d'entrées/sorties, des cellules des blocs logiques configurables (CLB) qui sont composés des interfaces d'entrées/sorties, des blocs logiques configurables (CLB), des multiplexeurs, des registres de décalage et des bascules (flip-flop), des interconnexions, des multiplexeurs et des mémoires SRAM.

## 2.3 Langage de description matérielle HDL

### 2.3.1 Évolution des langages HDL

La description matérielle a connu une évolution rapide qu'on peut résumer dans les quatre étapes suivantes.

Au début, la réalisation des composants (transistors) pour une logique combinatoire se faisait manuellement sur un papier spécial avec des crayons. On la nomme dessin au micron [HDL 13]. Une telle technique ne permet pas de réaliser des circuits très complexes.

L'évolution de la technologie de la microélectronique vers l'intégration de nombres plus importants du composant transistor a imposé de nouveaux outils de conception. C'est ainsi que les langages de description matérielle (HDL : *hardware description langages*) ont vu le jour. Ces langages sont des langages de description de haut niveau. En fait, un langage est dit de haut niveau lorsqu'il fait le plus possible l'abstraction de l'objet auquel ou pour lequel il est écrit. Le concepteur avec l'aide des outils informatiques de conception définit la structure du système, comme la manipulation des composants élémentaires de l'ordre de dizaines de transistors ainsi que les portes d'entrée d'un composant comme le PAL.

L'apparition des interfaces graphiques a facilité la mise en œuvre des circuits intégrés par schémas. En effet, il est généralement plus facile de lire et comprendre un schéma d'une description textuelle. On parle alors de la description schématique [HDL 13].

Enfin, les langages fonctionnels de description qui permettent non seulement la description à un niveau d'abstraction plus élevé, mais ont aussi répondu à des besoins fondamentaux des concepteurs de circuits intégrés :

- La réduction des temps de conception.
- L'accélération des simulations qui devenaient prohibitives avec l'accroissement de la complexité des systèmes.
- La normalisation des échanges fichier, le langage de description VHDL est devenu une norme IEEE numéro 1076 en 1987 [HDL 00]. Ce qui permet et facilite l'échange entre partenaires industriels, entre fournisseurs et clients.
- L'anticipation, la portabilité et la réutilisabilité, grâce aux modèles HDL il est possible de concevoir un système alors que ses composants ne sont pas

encore disponibles. Les langages normalisés sont très largement portables et les modifications et l'adaptation sont rendues plus simples donc moins risquées et moins coûteuses.

- La fiabilité de ces langages qui sont conçus et standardisés pour limiter en principe les risques d'erreur.

### 2.3.2 Utilité des langages HDL

Les langages HDL sont des langages portables qui vont trouver place dans le cycle de conception. Ils offrent deux avantages majeurs :

La simulation qui a pour but de vérifier le bon fonctionnement d'un système à implémenter. Ce modèle doit être précis dans son champ d'application et doit aussi être efficace. La simulation est réalisée par des bancs d'essai dont des stimuli sont associés aux signaux d'entrées du modèle ce qui permet l'analyse du temps de réaction de chaque signal du circuit du modèle. Dans l'industrie, le code VHDL de simulation qu'on nomme *testbench* joue un rôle très important, car il est intégré dans les spécifications d'un système.

Le deuxième avantage, c'est que les langages fonctionnels de descriptions matérielles servent à concevoir le modèle simulé dans une puce comme le circuit FPGA. Alors, il s'agit maintenant de ne plus modéliser en vue de simulation, mais pour décrire les composants digitaux qui seront implémentés.

### 2.3.3 Exemple de langage de description matérielle HDL

Il existe plusieurs langages HDL, les plus connus parmi eux sont les langages VHDL et Verilog.

#### 2.3.3.1 VHDL (very high speed integrated circuits hardware description language)

Le langage de description matérielle VHDL a été développé dans les années quatre-vingt au Portugal, il est ensuite devenu une norme IEEE numéro 1076 en 1987 [HDL 00]. Il est révisé en 1993 pour supprimer quelques ambiguïtés et améliorer la portabilité du langage, cette norme est vite devenue un standard en matière d'outils de description de fonctions logiques. À ce jour, on utilise le langage VHDL pour :

- concevoir des ASIC ;
- programmer des composants programmables de types PLD, CPLD et FPGA ;
- concevoir des modèles de simulations numériques ou des bancs d'essai.

Le flot de conception par le langage de description matérielle VHDL est représenté sur la Figure 2-6 :

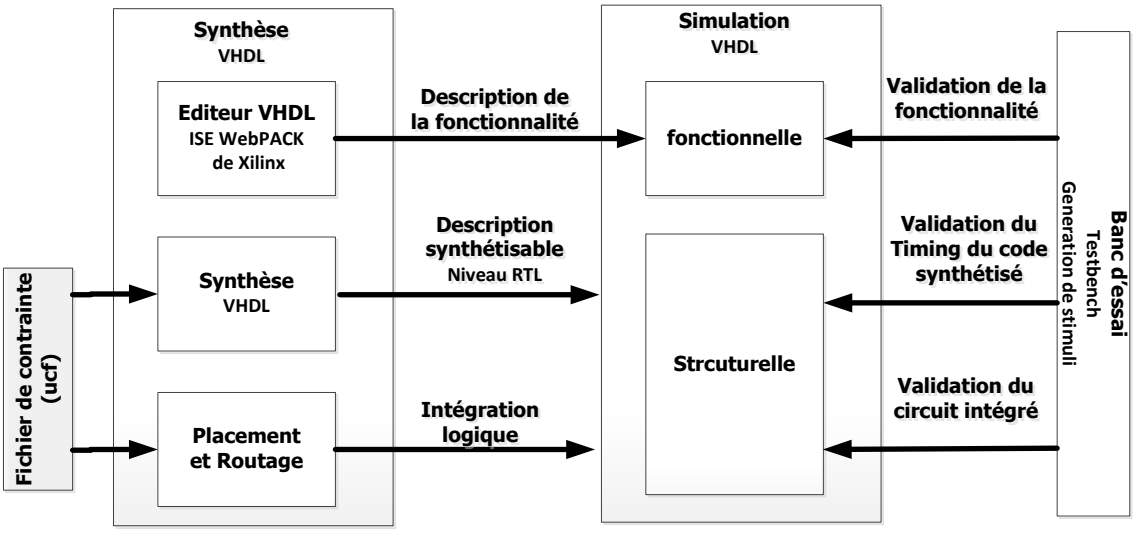



Figure 2-6 : Le flot de conception par le langage de description matérielle VHDL

#### 2.3.3.2 Langage de description matériel Verilog

Le langage de description matérielle Verilog a été développé aux alentours de l'année 1984 par la société Gateway Design Automation [VER 00]. Dans un premier temps, elle a conçu ce langage pour être utilisé dans leurs simulateurs logiques, mais avec le succès grandissant du langage, VHDL a mené ces concepteurs à faire de Verilog un standard ouvert, c'est le standard IEEE 1364 [NOR 00]. La syntaxe de Verilog est inspirée du langage de programmation C, bien que cette inspiration se limite au niveau d'expressions utilisées.

Avant de passer à la description du modèle de faute des circuits FPGA à base de mémoire statique, une présentation de l'architecture de la cellule d'une mémoire SRAM qui représente la majorité des éléments internes de FPGA, et leur modèle de faute est nécessaire.

## 2.4 Cellule de la mémoire statique SRAM et son modèle de faute

L'objectif de cette partie est de présenter l'architecture des FPGA à base SRAM et de mettre en évidence leurs spécificités. En effet, la compréhension de la problématique nécessite une description détaillée des fonctionnalités présentées au sein de ces composants complexes. Dans un premier temps, nous décrivons le fonctionnement d'une cellule mémoire SRAM et une partie de leur modèle de défaillance. L'architecture des FPGA à base SRAM sera ensuite présentée.

### 2.4.1 Cellule de la mémoire statique SRAM

Une cellule SRAM classique constitue le moteur du développement technologique dans l'industrie des semi-conducteurs et elle est composée de deux inverseurs rebouclés permettant de stocker un bit mémoire tant que la cellule est alimentée. L'information est écrite ou lue à partir de deux transistors d'accès.

Ces derniers sont commandés sur la grille par la ligne d'adresse WL (*word line*) et la donnée est chargée à partir des lignes de données. Les composants mémoires SRAM sont généralement composés de cellules-mémoire à six transistors (6T) (Figure 2-7).

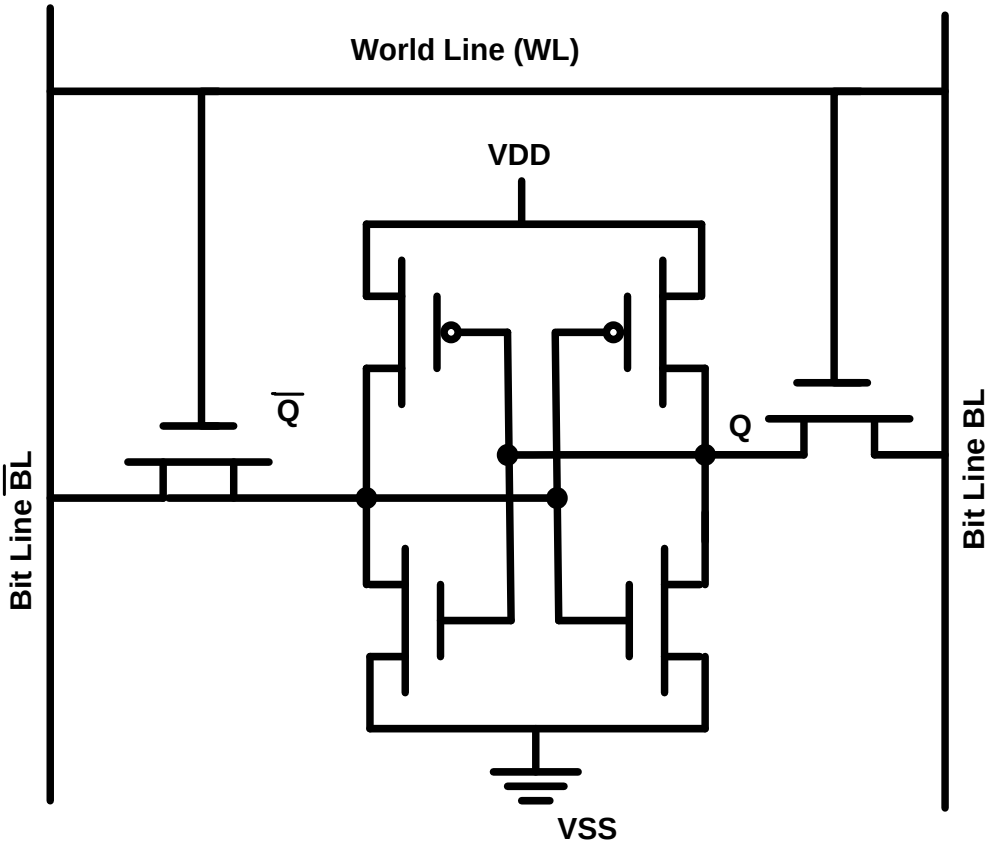



Figure 2-7 : La structure d'une cellule SRAM [AJG 87]

### 2.4.2 Défaillances, erreurs et fautes dans une cellule SRAM

La défaillance d'une cellule SRAM correspond à un fonctionnement incorrect de la cellule dû à des erreurs, ce qui constitue une différence entre une cellule saine et une cellule défaillante. Les fautes peuvent être permanentes ou non permanentes et sont causées notamment par une corrosion, une contamination ionique, un vieillissement, un alliage et des radiations, de même que des rayonnements cosmiques (Soft-Erros). Ainsi, la Figure 2-8 représente les différentes défaillances dans une cellule SRAM.

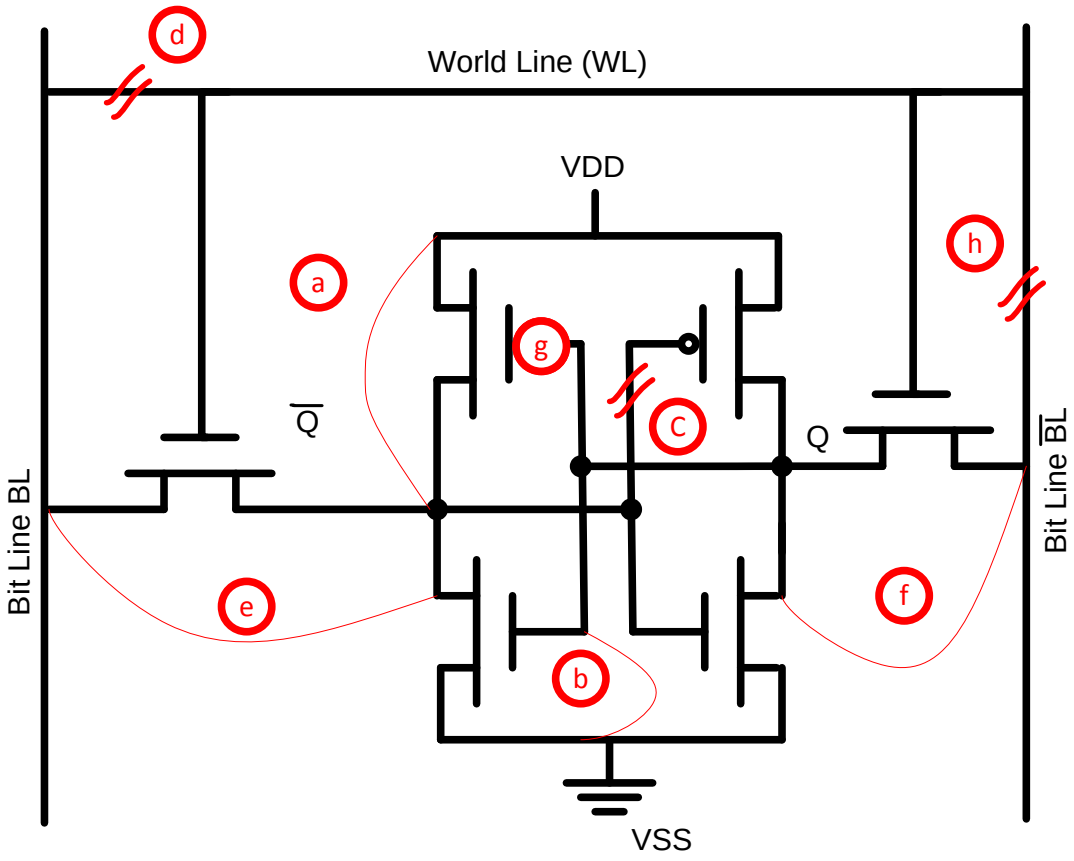



Figure 2-8 : Les défaillances dans une cellule SRAM

1. Une faute fonctionnelle causée par une faute de blocage *stuck-at* '1' (voir Figure 2-8 [a]).
2. Une faute fonctionnelle générée par le blocage de la cellule *stuck-at* '0' (voir Figure 2-8 [b]).
3. Une faute fonctionnelle générée par le blocage de la cellule *stuck-at* '0' (voir Figure 2-8 [c]).
4. Une faute fonctionnelle générée par le blocage de la cellule *stuck-at* '0' (voir Figure 2-8 [d]).
5. Ouverture de la ligne d'adresse (voir Figure 2-8 [e]).
6. Blocage de la ligne de données *stuck-at* '0' (voir Figure 2-8 [f]).
7. Blocage de la ligne de données *stuck-at* '1' (voir Figure 2-8 [g]).
8. La valeur '0' ne sera pas transmise sur la ligne de données BL (voir Figure 2-8 [h]).

### 2.4.3 Modèle de faute dans une cellule SRAM

Selon l'ouvrage [AJG 87], les fautes fonctionnelles dans les cellules SRAM peuvent être classées comme suit :

- les fautes de blocage SAF caractérisées par l'implication d'une seule cellule ;
- les fautes de transition TF caractérisées par l'implication d'une seule cellule ;

- les fautes de couplage CF dans deux ou plusieurs cellules sont impliquées ;
- les fautes de voisinage PSF dans plusieurs cellules sont impliquées.

#### 2.4.3.1 Fautes de blocage *stuck-at fault* (SAF)

Les fautes de blocage sont des fautes (Figure 2-9, Figure 2-10 et Figure 2-11) permanentes qui génèrent une valeur logique d'une cellule et sont toujours en zéro '0' (défaut SA0) ou bien un '1' (défaut SA1).

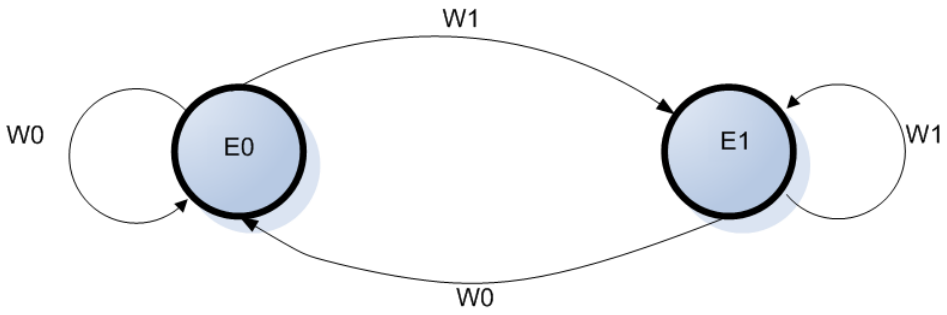



Figure 2-9 : Cellule saine [AJG 87]

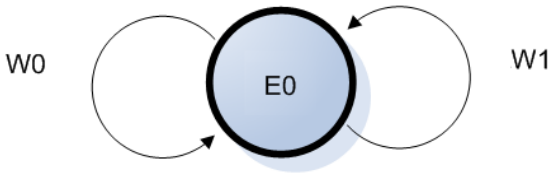



Figure 2-10 : Cellule affectée d'une SA0 [AJG 87]

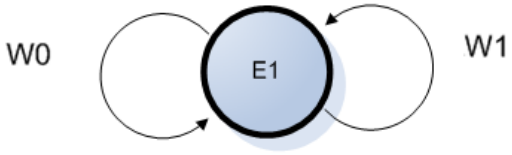



Figure 2-11 : Cellule affectée d'une SA1 [AJG 87]

#### 2.4.3.2 Fautes de transition TF

Les fautes de transition sont des fautes permanentes dont la valeur logique de la cellule SRAM ne peut pas effectuer la transition de zéro '0' vers '1' et vice versa. Les fautes de transition sont classées comme suit :

- Une cellule ne peut pas effectuer une transition de '0' vers '1' (*up transition fault* [0->1]) ;

- Une cellule ne peut pas effectuer une transition de ‘1’ vers ‘0’ (*down transition fault* [1 0]).

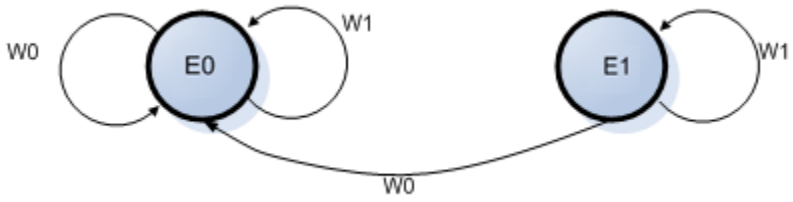



Figure 2-12 : Cellule affectée d’une TF *up transition fault* (w0 w1 r1) [AJG 87]

Les fautes causées par une *up transition* peuvent être détectées par l’écriture de la valeur zéro ‘0’ (en écrit ‘w0’) et un ‘1’ (en écrit ‘w1’) et la lecture de la valeur un ‘1’ (en écrit ‘r1’). L’algorithme est résumé comme suit : (w0 w1 r1).

#### 2.4.3.3 Fautes de couplages CF

Les fautes de couplage constituent une faute permanente dans le contenu d’une cellule. En cas de faute de couplage dépendant non seulement de l’état de la cellule, mais aussi de l’état d’autres cellules. Deux groupes peuvent être distingués :

- une faute de couplage dans le cas où deux cellules sont impliquées (une cellule victime et une cellule agresseur) ;
- une faute de couplage se produit lorsque 2xk- cellules sont impliquées.

#### 2.4.3.4 Fautes de voisinage *pattern sensitive* PSF

Le contenu de la cellule victime est forcé à 0 ou à 1 si un certain nombre de cellules voisines présentent une configuration particulière.
L’exécution de l’algorithme pour la détection des PSF ne peut s’effectuer que dans la phase du démarrage et peut prendre un délai bien considéré pour son exécution. Ce qui fait que, dans un système en marche, les fautes de voisinage sont difficiles à détecter, car, pour chaque cellule, l’effet de toutes les combinaisons possibles ($2^n$) des cellules adjacentes doit être testé. Tout cela rend la détection des erreurs aléatoires sans utilisation des méthodes précises presque impossible.

## 2.5 Modèle de faute du circuit FPGA

Un circuit FPGA comme déjà décrit dans le paragraphe précédent ne possède pas une structure unique (SRAM Based FPGA, les FPGA à structures hiérarchiques, etc.) ; de plus son architecture varie d’un fabricant à l’autre (Altera, Xilinx, Actel, Lucent, etc.) et dépend aussi de la façon par laquelle il va être programmé. Selon la littérature [JQI 04], il existe différentes fautes du matériel dans la puce FPGA.

### 2.5.1 Fautes de pontage (*briging fault*, BF)

La faute de pontage (BF) représente un court-circuit dans un groupe de lignes ou dans deux cellules. Ce genre de faute constitue une faute bidirectionnelle due à un niveau logique et non pas à une transition. Si les lignes ou les cellules prennent une valeur du ET de leur valeur saine, on parle de ET-Type et si elle prend la valeur de OU, on parle de OU-Type.

### 2.5.2 Fautes de blocage (*stuck-at fault*)

Les fautes de blocage sont des fautes permanentes et l'un des modes de défaillances significatifs qui se produisent dans les interconnexions dans un circuit FPGA. Cela dépend aussi de l'application pour détecter les fautes possibles. Le programmeur a la possibilité, au niveau RTL, d'injecter des fautes dans les registres de ce composant et d'observer les conséquences sur les résultats lors de la simulation.

### 2.5.3 Fautes de retard (*delay fault*, DF)

Les fautes de retard sont les défauts provoqués par des retards dans les composants combinatoires. Les interconnexions sont la principale source de grande variation du retard de propagation. Il est difficile de détecter ce genre de fautes ; la simulation avec des architectures dédiées à la sûreté de fonctionnement permet de diagnostiquer ce genre de faute.

### 2.5.4 Fautes d'interconnexions (*interconnect defect*)

Ce genre de défauts peut généralement être modélisé par les fautes de pontage (*bridging faults*) ou/et de fautes de blocage, qui se produisent dans les blocs logiques programmables CLB ainsi dans les entrées et les sorties de la puce FPGA.

## 2.6 Méthode de détection des erreurs dans un circuit FPGA

Les méthodes de détections des fautes aléatoires dans un circuit FPGA peuvent être classées en deux catégories [MJM 04].

- Test par configuration de la puce FPGA.
- Test par modification du matériel du FPGA pour le rendre testable, par l'insertion des chaînes de Boundary Scan par l'interface JTAG. Cette méthode est connue sous le nom conception en vue du test DFT (*design for testability*).

La plupart de ces méthodes décrites aussi dans [JQI 04] s'appliquent au test off-line, c'est-à-dire le FPGA n'est pas en service, qu'au test on-line.

### 2.6.1 Test de configuration de la puce FPGA

La détection des fautes citées ci-dessus par le test de configuration pour un circuit FPGA dont les cellules de configuration sont faites à partir de mémoire SRAM (SRAM-Based FPGA) utilise des vecteurs de test qui peuvent être appliqués sur la puce. Parmi

les tests de ce genre cités dans [JQI 04] et [MJM 04], c'est la méthode *built-in-self-test* (BIST), dont l'ensemble des CLB est divisé en deux parties, la première partie est sous test (BUT : *built under test*) alors que la deuxième partie est implémentée comme un générateur de vecteur de test (TPG : *test pattern generator*) et comme un analyseur de réponse de sortie (XOR).

Selon la littérature [MJM 04], le concept BIST peut être illustré par la Figure 2-13. Les composants nécessaires pour effectuer le test BIST sont d'une part un générateur de bit de parité et un générateur de test de vecteur de test automatique (ATPG : *automatic test pattern generator*), qui peut être réalisé par un compteur à N bits. L'analyse de réponse est accomplie en observant l'état du bit de parité qui sera toujours à '0' dans un circuit CLB défectueux.

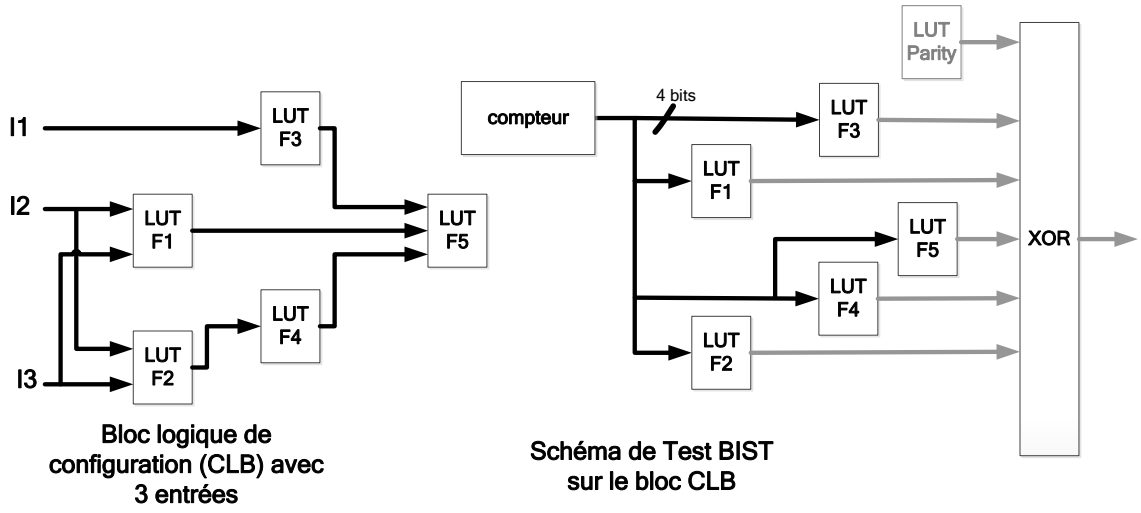



Figure 2-13 : Schéma du test BIST sur le bloc programmable logique CLB [MJM 04]

Cette stratégie proposée aussi par [MJM 04] ne nécessite que trois configurations de test pour tester les fautes de blocage dans les interconnexions ainsi dans les cellules CLB. Par ailleurs, en utilisant cette méthode de test, un diagnostic sur la puce FPGA peut être effectué très rapidement sans avoir besoin de configuration de tests supplémentaires. Ce genre de test nommé aussi mult-iconfiguration [MJM 04] ou bien la méthode des arbres ET/OR [JQI 04].

Dans la méthode des arbres ET/OR, toutes les LUT (*look-up-table*) intégrées dans une cellule CLB sont reconfigurées par la logique ET, et toutes les bascules utilisées dans l'application prennent la valeur '1'. Un vecteur de test d'une valeur '1' est généré vers l'entrée (INC_CLB) comme le montre la Figure 2-14. Une erreur de blocage à '0' survenue dans la configuration de l'interconnexion peut être détectée.

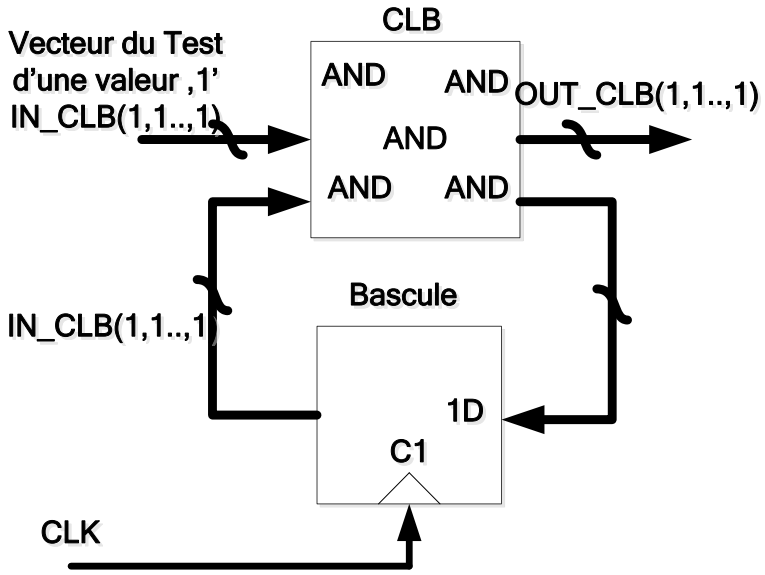



Figure 2-14 : Schéma de test des arbres ET/OU

La sortie OUT_CLB restera similaire à la valeur de l'entrée pendant des périodes de temps, s'il n'existe pas d'erreurs de matériel dans les interconnexions (faute de blocage à '0') dans les cellules CLB.

Par ces modèles de fautes de détection qui ont été présentés, les fautes de blocage à '0' ou '1', les fautes de court-circuit, les fautes de circuit ouvert pour les cellules des blocs logiques programmables CLB et des interconnexions peuvent être détectées.

Sachant que l'architecture des circuits FPGA est différente l'une de l'autre selon le fabricant, ce qui rend ces méthodes de tests spécifiques à un seul genre du circuit FPGA bien précis, car si on applique le test des arbres avec un autre type de FPGA on obtiendra des résultats différents, voire erronés.

### 2.6.2 Test par modification du matériel du FPGA

Une autre méthode de détection des erreurs aléatoires dans un circuit FPGA, c'est la méthode de scan modifiée [TKF 95] qui propose de tester non seulement les interconnexions, les blocs logiques programmables, mais aussi tous les éléments du FPGA. Les auteurs [ATH 99] présentent une approche qui permet de tester le FPGA, en le considérant comme un ensemble d'éléments indépendants à partir de la connexion des composants CLB sous forme d'une matrice unidimensionnelle (1-D). Ce qui rend tous genres de FPGA avec n'importe quelle grandeur testable.

Cette méthode a besoin également d'une légère modification de la mémoire SRAM de programmation du circuit FPGA, offrant la possibilité de décaler les données de configuration, et ainsi faire le test en chargeant en mémoire les données de configuration une seule fois, au lieu de le faire pour chaque séquence de test.

Cependant ces techniques reposent sur l'hypothèse que les SRAM sont construites à partir des registres, ce qui n'est pas toujours le cas. Il est impossible de décaler les données lorsque les SRAM sont des RAM classiques.

## 2.7 Conclusion

Les circuits FPGA sont puissants, flexibles, et offrent une disponibilité immédiate du matériel et cela est dû à la structure interne de la puce FPGA qui se compose de plusieurs petites unités de blocs logiques, également appelés blocs logiques programmables (CLB), lesquels, à leur tour, sont reliés entre eux par un réseau de connexions. Les unités possèdent également des blocs classiques de mémoire statique (SRAM) pour stocker les données, ainsi que des entrées et des sorties. Tout cela permet la réalisation des systèmes très complexes et performants.

Cependant, cette complexité matérielle provoque des défauts aléatoires qui doivent non seulement être détectés, mais aussi donner la possibilité de ramener le système à un état sûr en cas d'occurrence d'une défaillance aléatoire.

Un intérêt particulier a été accordé à la présentation et à la modélisation des différentes erreurs, fautes et défaillances, qui peuvent se produire dans une cellule mémoire de genre statique (SRAM). On a présenté aussi la structure interne des circuits FPGA ainsi que les différentes défaillances et les mécanismes de détection des fautes du matériel.

# Chapitre 3

# Terminologies de la sécurité fonctionnelle

## *Résumé du chapitre 3 :*

*Ce chapitre présente et définit les différentes terminologies nécessaires à la compréhension d'un système instrumenté de sécurité (SIS) : sécurité fonctionnelle, fiabilité, disponibilité, maintenabilité, sûreté de fonctionnement, taux de défaillance, niveau de performance requis, probabilité de défaillance dangereuse et niveau de sécurité. Il présente également la structure des différentes architectures dédiées à ces systèmes à savoir les architectures un parmi un (*1 out of 1*), un parmi un avec diagnostic un (*1 out of 1D*), un parmi deux (*1 out of 2*), un parmi deux avec diagnostic (*1 out of 2D*), deux parmi deux (*2 out of 2*), ainsi qu'une analyse détaillée des pannes qui peuvent affecter le moteur d'inférence floue avec sécurité avec les tests de diagnostic et les tests d'inspection associés. On présente également la modélisation de calcul de la valeur de la probabilité de défaillance dangereuse d'un système avec différentes architectures en utilisant trois méthodes : modélisation par des blocs-diagrammes de fiabilité, modélisation à partir de l'arbre des causes, méthode qui consiste à modéliser le comportement du système en présence des fautes possibles et enfin, modélisation grâce à des graphes de Markov, qui décrivent les états intermédiaires du système.*

# 3 Terminologies de la sécurité fonctionnelle

## 3.1 Introduction

L'évaluation quantitative du moteur d'inférence flou avec sécurité est basée sur une partie du processus de conception de la fonction de sécurité proposée par la norme CEI 61508 [CEI 06] dans le domaine de la sûreté de fonctionnement des systèmes instrumentés de sécurité (SIS) [FCI 11]. Ce processus est composé de plusieurs étapes :

- analyse et estimation du risque de la fonction de sécurité ;
- création de l'architecture du système ;
- détermination du taux de défaillance des composants, des sous-systèmes et du système ;
- détermination de la probabilité de défaillance à la demande PFD(t) sur l'intervalle du temps $[t_1, t_2]$.

Une partie d'évaluation quantitative est aussi basée sur une partie de la norme EN ISO 13849 [ISO 06] qui nous offre une méthode itérative pour évaluer les risques du système en utilisant un graphe de risque qui exige du concepteur de prédéfinir un niveau de performance requis (PLr : *required performance level*) de la fonction de sécurité qui devrait protéger une partie définie du système. Cette norme offre aussi une analyse probabiliste du risque résiduel d'un système et indique une corrélation entre le facteur de niveau de sécurité SIL (*safety integrity level*) défini dans la norme 61508 et le niveau de performance requis.

## 3.2 Paramètres caractéristiques des systèmes d'instrumentation de sécurité

La sûreté de fonctionnement a une importance capitale dans le cycle de vie de tout équipement destiné à réaliser une fonction bien précise. C'est le cas par exemple des équipements médicaux dans les unités de soins intensifs, ils doivent fonctionner d'une manière fiable et sûre comme les systèmes ou bien les équipements électroniques dans l'aéronautique. Les systèmes d'exploitation sûrs et fiables sont également une condition préalable décisive pour le bon fonctionnement d'une production industrielle complexe dont plusieurs processus doivent être contrôlés.

Les études de sûreté de fonctionnement utilisent un ensemble d'outils et de méthodes qui permettent, dans toutes les phases de vie d'un système, de s'assurer que celui-ci va accomplir les missions pour lesquelles il a été conçu, et cela dans le respect de la fiabilité, de la maintenabilité, de la disponibilité et de la sécurité prédéfinies. Ces études consistent généralement à analyser les effets des pannes, des dysfonctionnements, des erreurs d'utilisation ou des agressions du système à étudier.

Le terme sûreté de fonctionnement, on le trouve dans le plus haut niveau de la hiérarchie des systèmes de sécurité électroniques. Généralement, le terme sûreté de fonctionnement englobe les termes fiabilité, maintenabilité, disponibilité, et sécurité.

Dans la sûreté de fonctionnement, des défaillances au niveau de logiciels peuvent produire un fonctionnement incorrect du système et le mettre dans un état critique avec des conséquences graves. Pour cela le développement des équipements matériels hardware et software dédiés à des applications de la sécurité exigent un flot de conception spécifique.

Dans ce qui suit, nous détaillerons les concepts qui constituent la sûreté de fonctionnement à savoir la fiabilité, la disponibilité, la maintenabilité et la sécurité.

### 3.2.1 Fiabilité

La fiabilité est un terme générique qui couvre un large éventail de concepts et de mesures. Un système est fiable quand vous pouvez légitimement avoir une confiance sur la performance qu'il offre. L'approche globale de la fiabilité signifie également que les systèmes (par exemple les téléphériques, les avions, les automates de sécurité ou les trains à grande vitesse) conçus et construits doivent être soumis antérieurement à une vérification à certains intervalles du temps pour détecter les risques de sécurité. L'homme est considéré dans cette chaîne comme un facteur non négligeable qui peut générer des défaillances dangereuses souvent avec des conséquences catastrophiques. Il est donc très important d'identifier ces défaillances dans l'analyse de la fiabilité d'un système et de les prévoir.

La fiabilité d'un système est donc la probabilité du fonctionnement sûr d'un système pendant un certain temps précis selon son cahier des charges.

Si un système possède un taux de défaillance constant λ [défaillance/heure], la fiabilité d'un système au moment t est définie comme suit [JBÖ 07] :

$$R(t) = e^{-\lambda t} \text{ avec } t \quad en \ [h] \tag{3-1}$$

### 3.2.2 Disponibilité

La disponibilité est la probabilité de satisfaction du système, certaines exigences bien définies dans un délai aussi bien défini, on peut dire qu'il s'agit ici d'un critère de qualité du système. Afin de garantir un minimum de la disponibilité des systèmes standards, on doit faire une analyse statique des défaillances déjà connues, par conséquent on peut par exemple faire joindre à un sous-système une redondance afin de minimiser la probabilité de défaillance totale.

Ainsi, la disponibilité A est une fonction dépendante du temps t. Pour les systèmes ayant une défaillance et un taux de réparation constant, il existe une relation entre la valeur MTTF (temps moyen qui s'écoule jusqu'à ce qu'un système tombe en panne), MTBF (temps moyen entre deux défaillances) et la disponibilité (A) selon la relation [JBÖ 07] :

$$A = \frac{MTTF}{MTBF} \quad (3\text{-}2)$$

### 3.2.3 Maintenabilité

La maintenance est la durée nécessaire pour qu'un système tombant en panne revienne à l'état de fonctionnement.

### 3.2.4 Sûreté

La sûreté est la résistance du système envers les défaillances critiques. Une défaillance dangereuse est nocive aussi bien qu'une défaillance non critique (non dangereuse). Une défaillance dangereuse pourrait constituer un danger sérieux pour les personnes, le matériel et l'environnement. Le coût d'une telle défaillance peut être de plusieurs ordres de grandeur plus élevés que les coûts d'exploitation.

La réalisation d'un système qui répond aux critères de sûreté de fonctionnement, nécessite le déploiement des systèmes embarqués dans plusieurs applications critiques, telles que l'ABS ou bien le système airbag des véhicules, sa certification est faite par un organisme de certification indépendant, comme la société TÜV Rein-Neckar au Portugal.

### 3.2.5 Facteur de sécurité

Le facteur de sécurité S indique le nombre de pannes qui mène à la perte de la fonction de sécurité. Il est donné en pourcentage.

On distingue à partir du facteur de sécurité S deux genres de types de composants [JBÖ 07] :

1. les composants de type B qui possèdent une valeur de sécurité de S = 50 % ;
2. les composants de type A qui possèdent une valeur de sécurité de S = 10 %.

Les composants de type A se caractérisent par un mode de défaillance très bien défini et ces défaillances sont connues. On possède toujours pour ces composants un retour d'expérience déjà accumulée. L'afficheur, le transistor, la résistance le condensateur, etc., appartiennent tous à des composants de type A, car leur mode de défaillance est connu.

Par contre les composants de type B se caractérisent par des modes de défaillance qui ne sont pas tous définis, la testabilité n'est pas de 100 % et la pertinence de la valeur des données relatives au retour d'expérience est faible. Le facteur de sécurité pour les composants du type B est égal à S = 50 %. Pour ces composants une « analyse des modes de défaillance, de leurs effets et de leur criticité » — AMDEC est la traduction de l'anglais FMECA (*failure modes, effects and criticality analysis* — doit être effectuée.

Le facteur de sécurité S peut également être démontré explicitement via une analyse AMDEC pour les composants de type B. Pour les composants dont la sécurité n'est pas critique tels qu'un afficheur, il est supposé d'une valeur égale à S = 0 %.

### 3.2.6 Taux de défaillance

Un système peut subir une défaillance, cette défaillance peut être sûre ou bien dangereuse [FCI 11] (Tableau 3-1). Les défaillances qui ne causent pas un état critique du système ou bien qui n'affectent pas la fonction de sécurité sont considérées comme des défaillances sûres.

| **Genre de défaillance** | **Défaillance non dangereuse (sûre)** | **Défaillance dangereuse** |
|---|---|---|
| Défaillance détectée | Défaillance sûre détectée.<br>Le taux de défaillances est représenté par $\lambda^{SD}$ . | Défaillance dangereuse détectée.<br>Le taux de défaillances est représenté par $\lambda^{DD}$ . |
| Défaillance non détectée | Défaillance sûre non détectée.<br>Le taux de défaillances est représenté par $\lambda^{SU}$ . | Défaillance non dangereuse détectée.<br>Le taux de défaillances est représenté par $\lambda^{DU}$ . |

Tableau 3-1 : Genre de défaillance

Les différents taux de défaillances sont calculés à partir de la défaillance de base donnée par le fabricant. Elle est représentée par l'unité [FIT] (*failure in time*) qui indique la probabilité du temps d'échecs par heure [JBÖ 07].

$$FIT = 1*10^{-9} 1/h \tag{3-3}$$

On obtient les équations suivantes (Tableau 3-2) correspondant aux différents taux de défaillance d'un composant [JBÖ 07] :

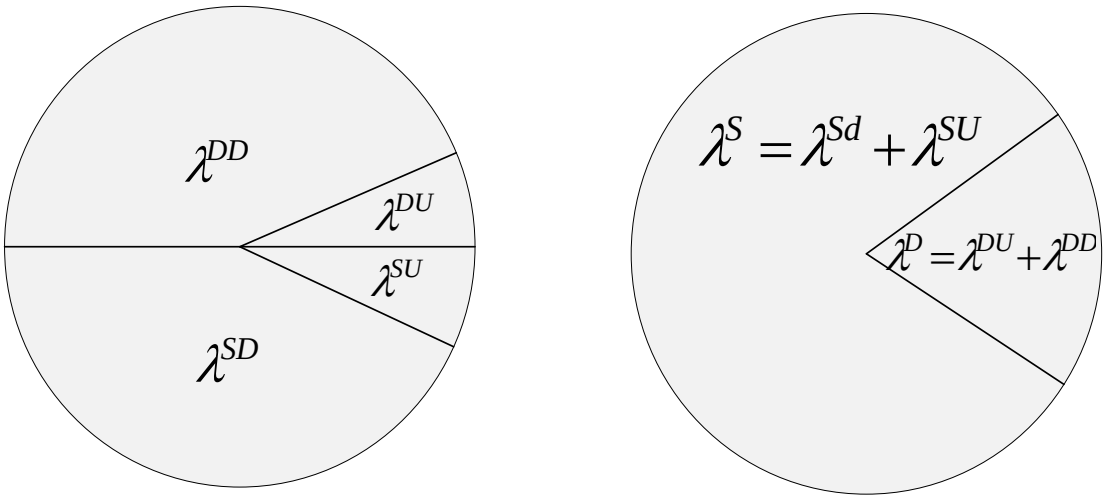
| Genre de taux de défaillance | Formule de calcul |
|---|---|
| Taux de défaillance dangereuse détectée | $\lambda^{DD} = \frac{\lambda}{2} DC$ |
| Taux de défaillance dangereuse non détectée | $\lambda^{DU} = \frac{\lambda}{2}(1 - DC)$ |
| Taux de défaillance sûre détectée | $\lambda^{SD} = \frac{\lambda}{2} DC$ |

Tableau 3-2 : Les équations du taux de défaillance

Considérons [FCI 11] une vanne utilisée dans un système instrumenté de sécurité pour couper le flux de produit lors de la sollicitation de la fonction de sécurité.

- La défaillance sûre correspond au blocage de la vanne en position fermée sans que la fonction de sécurité n'ait été sollicitée. Si ce blocage peut être détecté (par des tests de diagnostic) avant la sollicitation de la fonction de sécurité, il sera donc classé comme une défaillance sûre détectée, sinon elle sera classée en défaillance sûre non détectée.
- La défaillance dangereuse correspond au blocage de la vanne en position ouverte. Si ce blocage peut être détecté (par des tests de diagnostic) avant la sollicitation de la fonction de sécurité, il sera donc classé comme une défaillance dangereuse détectée, sinon elle sera classée comme une défaillance dangereuse non détectée.

Le comportement des composants non programmables de sécurité est très bien connu puisque leur structure n'est pas complexe. Les défaillances sûres (détectées et non détectées) représentent la plus grande proportion (Figure 3-1 [FCI 11]), par contre les composants programmables de sécurité, leurs défaillances sûres (détectées et non détectées) représentent seulement la moitié. Cela est dû à la complexité de fabrication de ces derniers.




a) Composant programmable b)Composant non programmable

Figure 3-1 : Défaillances d'un composant programmable et non programmable.

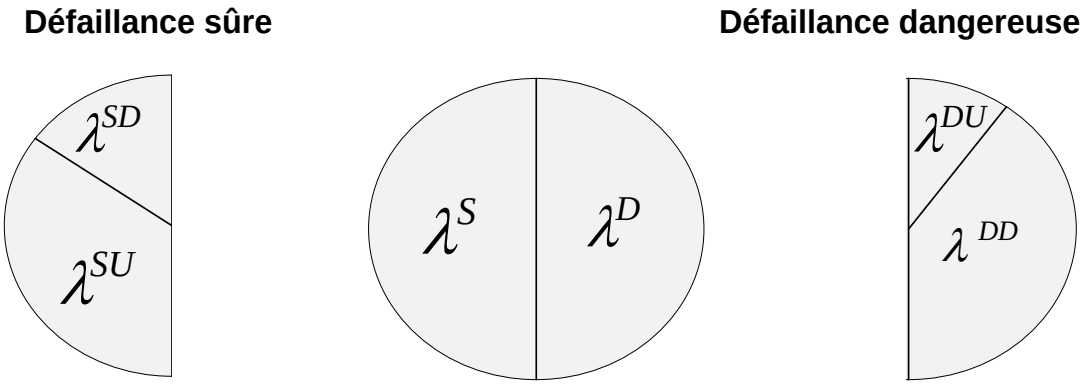



Figure 3-2 : Les types des défaillances d'un système programmable

### 3.2.7 Architecture d'un système instrumenté de sécurité (SIS)

Pour caractériser l'architecture d'un système avec sécurité, la convention MooN sera utilisée, ce qui signifie que M canaux sur les N canaux que comporte le système doivent fonctionner correctement pour que la fonction de sécurité soit exécutée.

Par exemple, un système d'architecture 1ooN dédié à la sécurité, ou sous-ensemble d'un tel système, constitué de N canaux indépendants qui sont connectés de telle sorte qu'il suffit qu'un seul canal (M = 1) soit opérationnel pour que la fonction de sécurité soit assurée. Par contre, pour les systèmes de structure 2ooN, l'assurance de la

fonction de sécurité exige que les deux (M=2) canaux sur les N canaux soit opérationnels.

Dans les applications industrielles citées dans les ouvrages [GBL 10] et [JBÖ 07], on trouve principalement les architectures suivantes :

- 1oo1 (*1 out of 1*) un parmi un ;
- 1oo1D (*1 out of 1 with diagnostic*) un parmi un avec diagnostic ;
- 1oo2 (*1 out of2*) au moins un parmi deux ;
- 1oo2D (*1 out of 2 with diagnostic*) au moins un parmi deux avec diagnostic ;
- 2oo2 (*2 out of 2*) deux parmi deux ;
- 2oo2 (*2 out of 2 with diagnostic*) deux parmi deux avec diagnostic.

#### 3.2.7.1 Architecture un parmi un 1oo1

Un système d'architecture « 1oo1 » (Figure 3-3) consiste en un seul canal, pour lequel une défaillance dangereuse entraînera la perte de la fonction de sécurité en cas de demande. Les tests de diagnostic sont présents ici pour assurer une détection de fautes en vue de réparer le système, mais n'affectent pas la sortie.
L'exemple suivant représente un système instrumenté de sécurité (SIS) de type système d'arrêt d'urgence de sécurité (ESD : *emergency safety shudown*) comprenant un capteur, connecté en série avec une logique de commande, qui commande un relais. Par définition, la fonction de sécurité se résume à l'ouverture de contact du relais en cas d'une détection d'une défaillance dangereuse au niveau du canal.

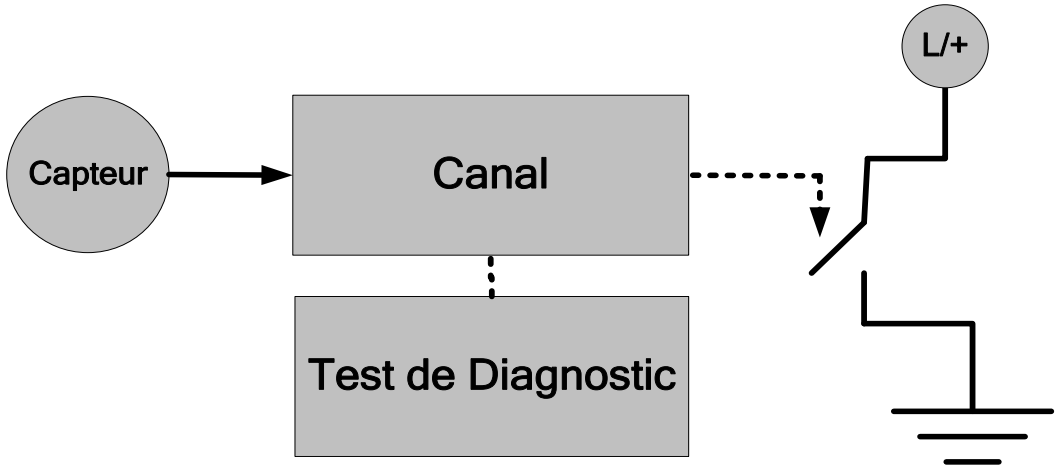



Figure 3-3 : L'architecture 1oo1 du SIS de type ESD

Le Tableau 3-3 montre les différents états du système de structure 1oo1. Par définition, la fonction de sécurité se résume à l'ouverture de contact du relais en cas d'une détection d'une défaillance dangereuse au niveau du canal.

| La fonction de sécurité est assurée si le relais est ouvert | | | |
|---|---|---|---|
| **État du système** | **Causes potentielles** | **Disponibilité du système** | **Sécurité du système** |
| Canal défaillant | Relais bloqué à l'état fermé | Impossibilité d'actionner la sortie. (pas de disponibilité) | Le système n'est plus sûr |
| | Relais bloqué à l'état ouvert | Impossibilité d'actionner la sortie (pas de disponibilité) | Le système est sûr |

Tableau 3-3 : La disponibilité et la sécurité dans une structure 1oo1

### 3.2.7.2 Architecture un parmi un 1oo1D

Un système d'architecture un parmi un avec diagnostic (1oo1D) consiste en un seul canal, pour lequel une défaillance dangereuse entraînera la perte de la fonction de sécurité en cas de demande. Les tests de diagnostics sont capables de couper l'énergie des sorties en cas de détection d'erreur. On a amélioré la structure d'une architecture simple à une structure dont les tests de diagnostic agissent directement sur la sortie si les tests détectent une défaillance dangereuse. Il faut noter que les fautes dangereuses non détectées par les tests font passer le système de structure 1oo1D dans un état dangereux, car la fonction de sécurité ne peut pas être exécutée. La disponibilité d'exécution de la fonction de sécurité n'est donc pas satisfaite.

La Figure 3-4 représente un système instrumenté de sécurité de type de système d'arrêt d'urgence de sécurité (ESD : *emergency safety shudown*) comprenant un capteur, connecté en série avec une logique de commande, qui commande un relais et dont les tests de diagnostic agissent directement sur la sortie.

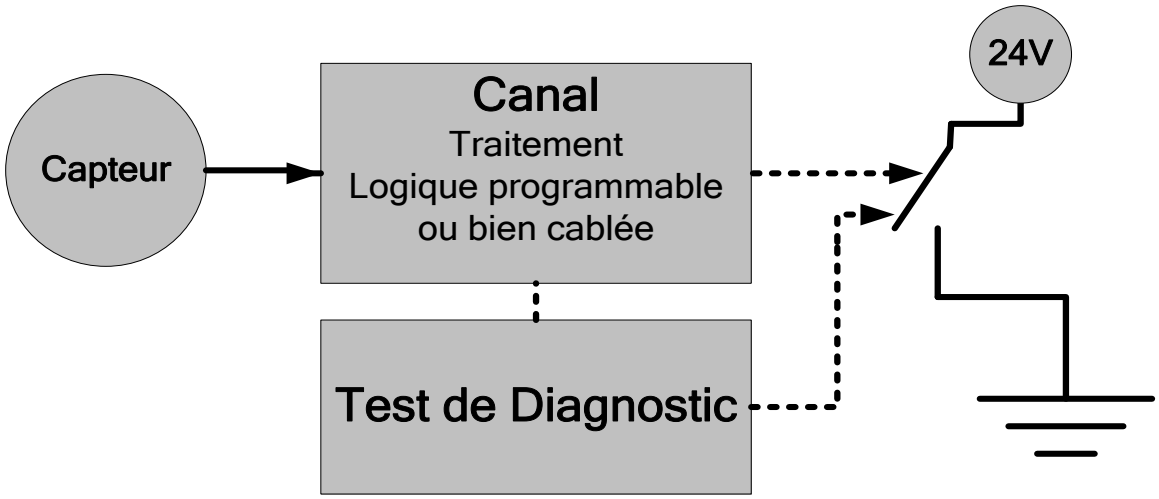



Figure 3-4 : L'architecture 1oo1D du SIS de type ESD

Le Tableau 3-4 montre les différents états du système 1oo1D. Par définition, la fonction de sécurité se résume à l'ouverture de contact du relais en cas d'une détection d'une défaillance dangereuse au niveau du canal.

<table>
<tr><th colspan="4">La fonction de sécurité est assurée si le relais est ouvert</th></tr>
<tr><th>État du système</th><th>Causes potentielles</th><th>Disponibilité du système</th><th>Sécurité de système</th></tr>
<tr><td rowspan="4">Canal défaillant</td><td rowspan="2">Relais bloqué à l'état fermé</td><td><u>Défaillance détectée</u><br>Possibilité d'actionner la sortie par les tests de diagnostic (disponibilité)</td><td>Le système est sûr</td></tr>
<tr><td><u>Défaillance non détectée</u><br>Impossibilité d'actionner la sortie par les tests de diagnostic (pas de disponibilité)</td><td>Le système n'est plus sûr</td></tr>
<tr><td rowspan="2">Relais bloqué à l'état ouvert</td><td><u>Défaillance détectée</u><br>Possibilité d'actionner la sortie par les tests de diagnostic (disponibilité)</td><td rowspan="2">Le système est sûr puisque la sortie n'est pas alimentée</td></tr>
<tr><td><u>Défaillance non détectée</u><br>Impossibilité d'actionner la sortie par les tests de diagnostic (pas de disponibilité)</td></tr>
</table>

Tableau 3-4 : La disponibilité et la sécurité dans une structure 1oo1D

### 3.2.7.3 Architecture au moins un parmi deux 1oo2

Un système d'architecture « 1oo2 » consiste en deux canaux, pour lequel une défaillance d'un seul des deux canaux n'empêche pas l'exécution de la fonction de sécurité. Le système est défaillant si les deux canaux ne fonctionnent pas. Les tests de diagnostic signalent les fautes pour les réparer, mais n'affectent pas la sortie.

La Figure 3-5 représente un système ESD instrumenté de sécurité avec une architecture 1oo2, comprenant deux capteurs, connectés en série avec deux logiques de commande, qui commandent deux relais.

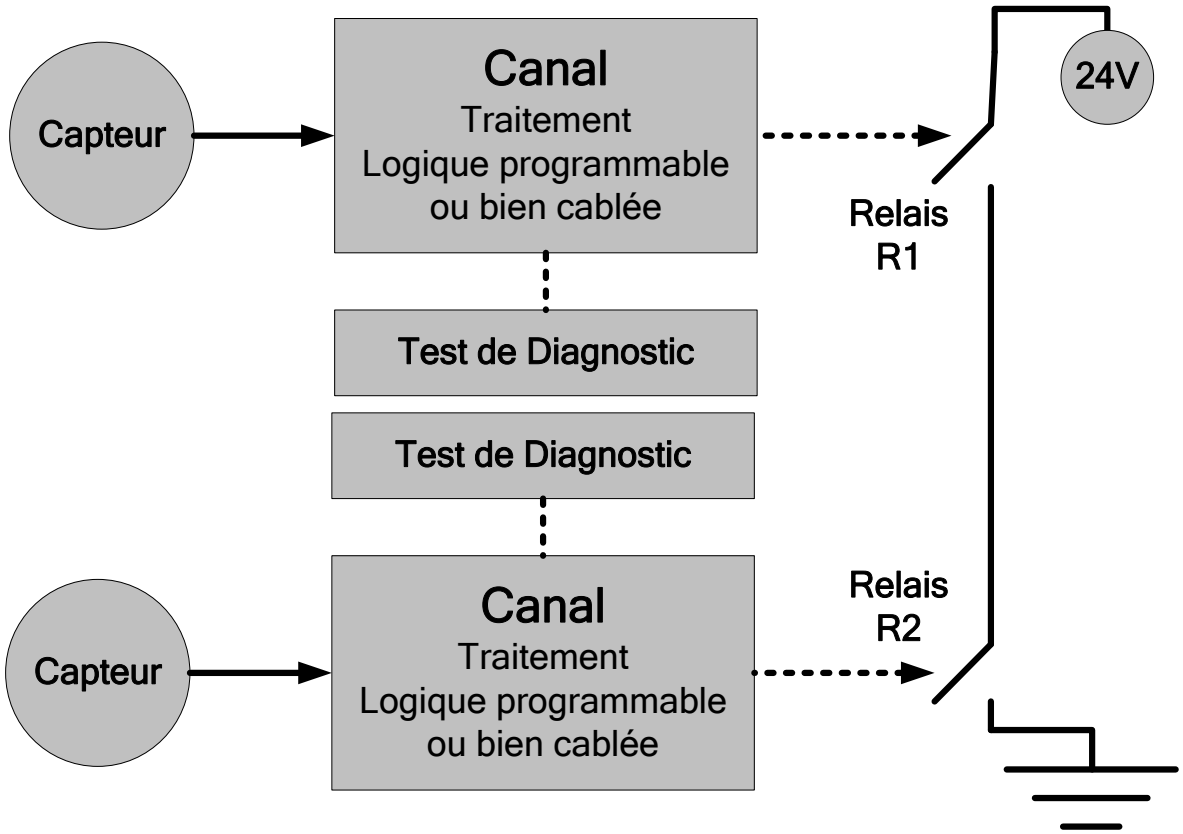



Figure 3-5 : L'architecture 1oo2 du SIS de type ESD

Le Tableau 3-5 montre les différents états du système 1oo2. Par définition, la fonction de sécurité se résume à l'ouverture de contact du relais en cas d'une détection d'une défaillance dangereuse.

| La fonction de sécurité est assurée si le relais est ouvert | | | |
|---|---|---|---|
| **État du système** | **Cause potentielle** | **Disponibilité du système** | **Sécurité de système** |
| Un canal est défaillant | Relais bloqué à l'état fermé | Possibilité d'actionner la sortie par le canal non défaillant. La disponibilité de continuer d'assurer la fonction de sécurité est satisfaite. | Le système est sûr, puisque la fonction de sécurité peut être assurée. |
| | Relais bloqué à l'état ouvert | Impossibilité d'actionner la sortie, par le canal non défaillant. | Le système est sûr puisque la sortie n'est pas alimentée. |
| Deux canaux sont défaillants | Relais bloqué à l'état fermé | Impossibilité d'actionner la sortie. | Le système n'est plus sûr. |
| | Relais bloqué à l'état ouvert | Impossibilité d'actionner la sortie. | Le système est sûr puisque la sortie reste non alimentée. |

Tableau 3-5 : La disponibilité et la sécurité dans une architecture 1oo2

#### 3.2.7.4 Architecture au moins un parmi deux 1oo2D avec diagnostic

Un système d'architecture « 1oo2 D » avec diagnostic consiste en deux canaux, pour lequel une défaillance d'un seul des deux canaux n'empêche pas l'exécution de la fonction de sécurité, on parle donc d'une tolérance aux anomalies de matériel HFT = 1 (*hardware failure tolerant*). Le système est défaillant lorsqu'aucun des deux canaux ne fonctionne. Les tests de diagnostics sont capables de couper l'énergie des sorties dès qu'une erreur aura été détectée.

L'exemple suivant représente un système instrumenté de sécurité de type de système d'arrêt d'urgence de sécurité (ESD : *emergency safety shudown*) comprenant deux capteurs, connectés en série avec deux logiques de commande, qui commandent deux relais.

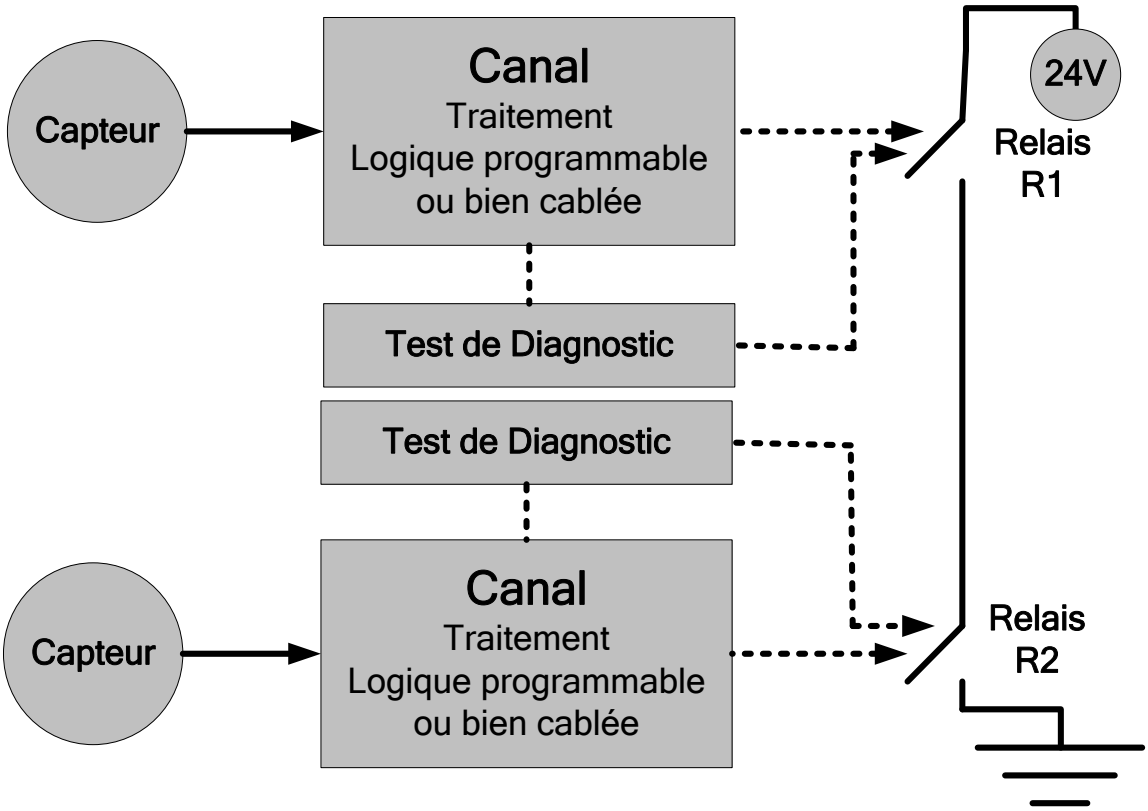



Figure 3-6 : L'architecture 1oo2D avec diagnostic du SIS de type ESD

Le Tableau 3-6 montre les différents états du système 1oo2D. Par définition la fonction de sécurité se résume à l'ouverture de contact du relais en cas d'une détection d'une défaillance dangereuse.

<table>
<tr><th colspan="4">La fonction de sécurité est assurée si le relais est ouvert</th></tr>
<tr><th>État du système</th><th>Causes potentielles</th><th>Disponibilité du système</th><th>Sécurité de système</th></tr>
<tr><td rowspan="2">Un canal est défaillant</td><td>Relais bloqué à l'état fermé</td><td><u>Défaillance détectée et non détectée</u><br>Possibilité d'actionner la sortie par le canal non défaillant. La disponibilité de continuer à assurer la fonction de sécurité est satisfaite.</td><td><u>Défaillance détectée et non détectée</u><br>Le système est sûr, puisque la fonction de sécurité peut être assurée.</td></tr>
<tr><td>Relais bloqué à l'état ouvert</td><td><u>Défaillance détectée et non détectée</u><br>Impossibilité d'actionner la sortie, puisque la sortie est toujours à l'état non alimenté</td><td><u>Défaillance détectée et non détectée</u><br>Le système est sûr, puisque la sortie n'est pas alimentée.</td></tr>
<tr><td rowspan="3">Deux canaux sont défaillants</td><td rowspan="2">Relais bloqué à l'état fermé</td><td><u>Défaillance détectée</u><br>Possibilité d'actionner la sortie.</td><td>Le système est sûr. La possibilité d'agir sur la sortie par les tests de diagnostic.</td></tr>
<tr><td><u>Défaillance non détectée</u><br>Impossibilité d'actionner la sortie.</td><td>Le système n'est plus sûr.</td></tr>
<tr><td rowspan="2">Relais bloqué à l'état ouvert</td><td><u>Défaillance détectée</u><br>Possibilité d'actionner la sortie.</td><td rowspan="2">Le système est sûr puisque la sortie reste non alimentée.</td></tr>
<tr><td><u>Défaillance non détectée</u><br>Impossibilité d'actionner la sortie.</td></tr>
</table>

Tableau 3-6 : La disponibilité et la sécurité dans une architecture 1oo2D

### 3.2.7.5 Architecture deux un parmi deux 2oo2

Un système d'architecture « 2oo2 » consiste en deux canaux pour lequel la défaillance d'un seul des deux canaux empêche l'exécution de la fonction de sécurité. La fonction de sécurité sera traitée si les deux canaux fonctionnent correctement. On parle donc d'une tolérance aux anomalies de matériel HFT = 0 (*hardware failure tolerant*).

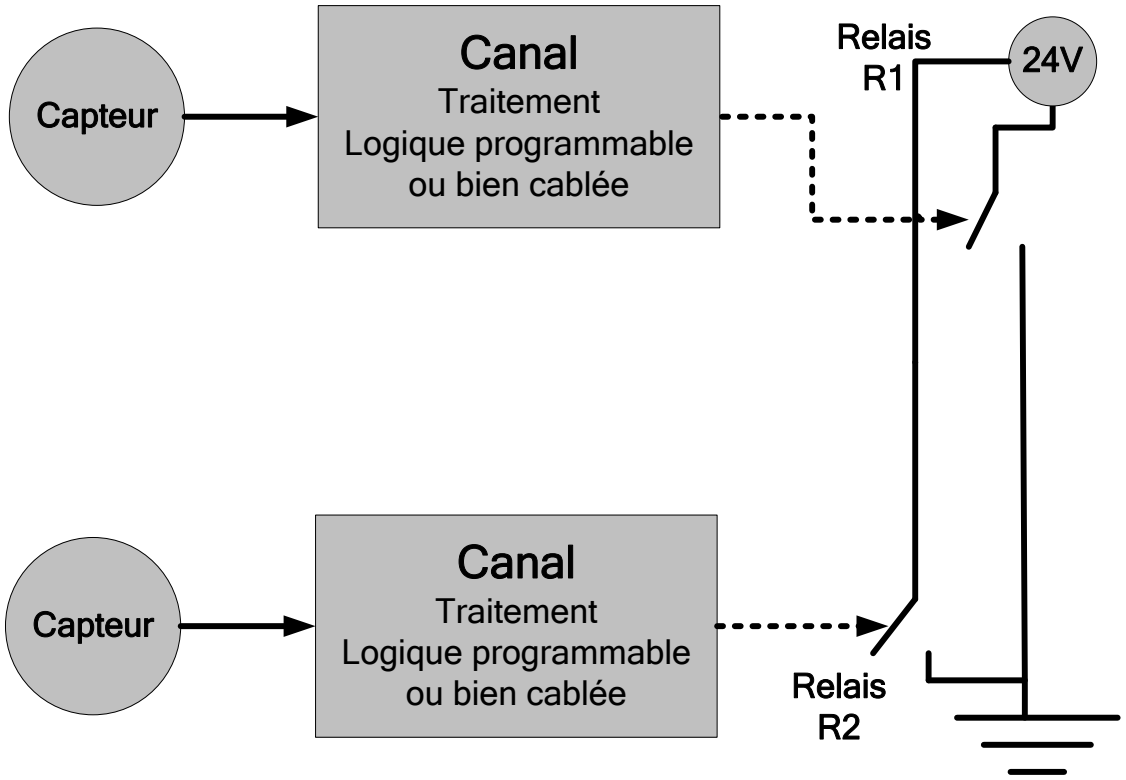



Figure 3-7 : L'architecture 2oo2 du SIS de type ESD

Le Tableau 3-7 montre les différents états du système d'architecture 2oo2

| **La fonction de sécurité est assurée si le relais est ouvert** | | | |
|---|---|---|---|
| **État du système** | **Causes potentielles** | **Disponibilité du système** | **Sécurité de système** |
| Un canal est défaillant | Relais bloqué à l'état fermé | Impossibilité d'actionner la sortie par le canal non défaillant. La sortie reste toujours alimentée. | Le système n'est plus sûr, la sortie est toujours alimentée. |
| | Relais bloqué à l'état ouvert | Possibilité d'actionner la sortie. | Le système est sûr puisque la sortie n'est pas alimentée. |
| Deux canaux sont défaillants | Relais bloqué à l'état fermé | Impossibilité d'actionner la sortie. | Le système n'est plus sûr. |
| | Relais bloqué à l'état ouvert | Impossibilité d'actionner la sortie. | Le système est sûr puisque la sortie reste non alimentée. |

Tableau 3-7 : La disponibilité et la sécurité dans une architecture 2oo2

Par définition, la fonction de sécurité se résume à l'ouverture de contact du relais en cas de détection d'une défaillance dangereuse.

### 3.2.8 Probabilité de défaillance dangereuse sur demande (PFD)

Il existe deux genres de types de cause de défaillances d'un système de sécurité.

[FCI 07]. Une défaillance peut, dans un premier cas, être due à des pannes aléatoires liées uniquement au matériel, résultant des divers mécanismes de dégradation et dont l'instant exact d'occurrence n'est pas prévisible. Dans un second cas, la défaillance du système peut être due à des pannes systématiques, pouvant par exemple être liées à des problèmes au niveau de la documentation, de la communication ou encore du traitement, etc.

### 3.2.9 Test de diagnostic

Les tests de diagnostic, en ligne, permettent de détecter des défauts de matériel, par exemple les tests de mémoire (Watchdog Test, Walking Bit Test, RAM Test) qui permettent la détection des erreurs aléatoires. Les tests de diagnostic agissent au niveau du composant et non au niveau de la fonction. Tandis que les tests périodiques, que l'on appelle aussi « tests d'inspection » (*proof test*) sont des tests périodiques hors ligne permettant de détecter des pannes dans le système, de sorte que celui-ci puisse être réparé et revenir à un état équivalent à son état initial.

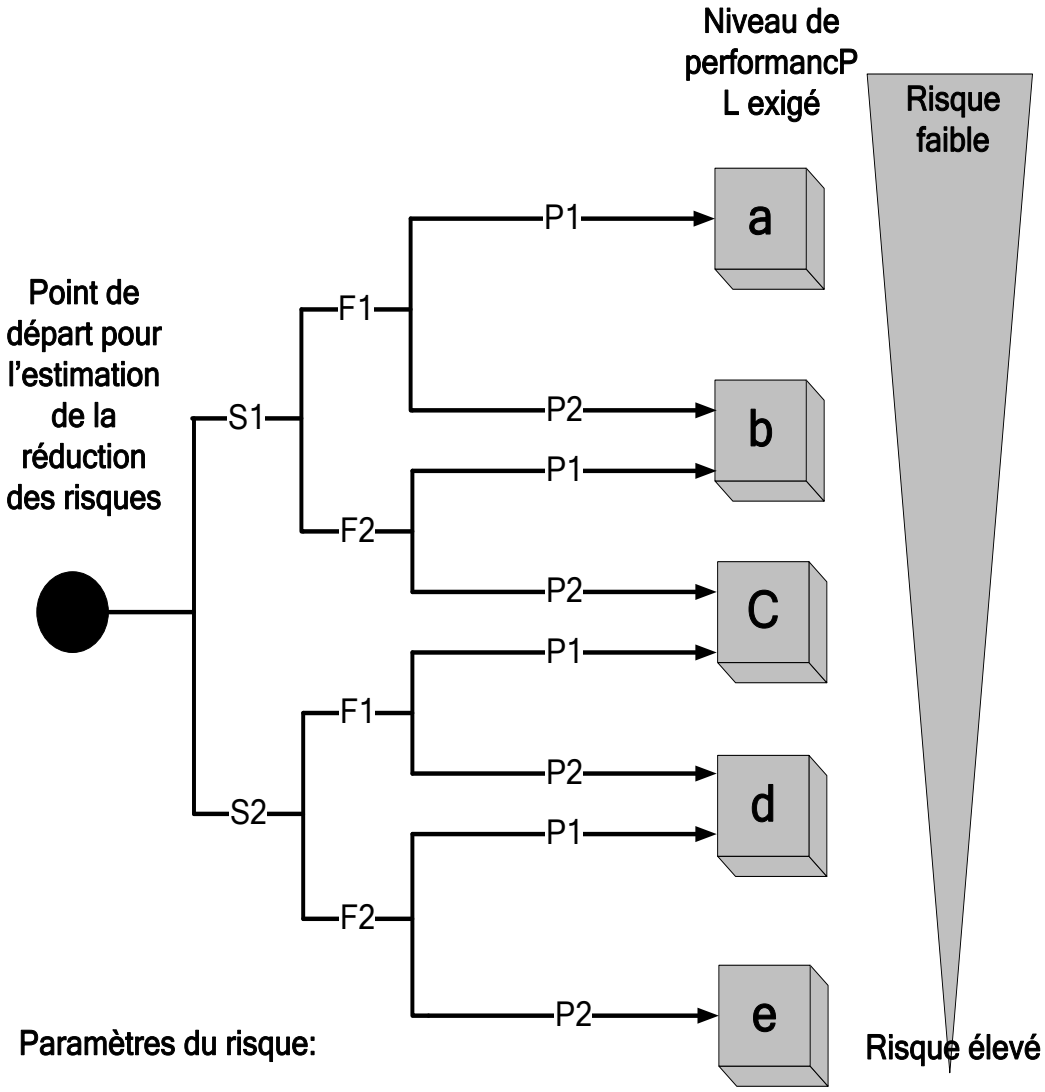
### 3.2.10 Niveau de performance PLr (*required performance level*)

La norme EN ISO 13849 [ISO 06] offre au constructeur une méthode itérative pour évaluer si les risques d'une machine ou d'un système peuvent être limités à un niveau résiduel acceptable par l'utilisation des fonctions de sécurité appropriées. La méthode adoptée prévoit, pour chaque risque, un cycle d'hypothèse-analyse-validation à l'issue duquel on doit être en mesure de démontrer que chaque fonction de sécurité choisie est adaptée au risque relatif à l'examen.

La première étape consiste donc à évaluer le niveau de performance requis pour chaque fonction de sécurité.

Selon EN ISO 13849-1 [ISO 02] à partir du point de départ (Figure 3-8), le constructeur de la machine ou le concepteur du système identifiera, en répondant aux questions S, F et P, le niveau de performance de la machine et la fonction de sécurité qui sera examinée.

Le niveau de performance requis est classé en cinq niveaux [ISO 02], en fonction de l'augmentation du risque et allant du niveau de performance *Pla* au niveau de performance *Ple*. Chacun d'eux identifie un domaine numérique de probabilité moyenne de défaillance dangereuse par heure (Tableau 3-8). Un niveau de performance *Pla* indique par exemple que la probabilité moyenne de défaillance dangereuse par heure est comprise entre $10^{-5}$ et $10^{-4}$.




S : gravité de la blessure

S1 : blessure légère (essentiellement rèversible)

S2 : blessure grave (généralement irrèversible y compris la mort)

F : frèquence ou/et durèe d'exposition (au danger)

F1 : de rare à assez frèquente et/ou court durèe

F2 : de frèquente à continue et/ou de longue durèe

P : possibilité de reduction du danger ou limitation du dommange

P1 : possible sous certaines conditions

P2 : Rarement possible

a,b,c,d et e sont les objectifs du niveau de performance relatifs à la sécurité

Figure 3-8 : Niveau de performance requis PLr

| Niveaux de sécurité | Niveau de performance PL | Probabilité moyenne de défaillances dangereuses par heure PFH (1/h) |
|---|---|---|
| - | a | $10^{-5} \leq \text{PFH} < 10^{-4}$ |
| SIL1 | b | $10^{-6} \leq \text{PFH} \leq 10^{-5}$ |
| SIL2 | c | $10^{-7} \leq \text{PFH} \leq 10^{-6}$ |
| SIL3 | d | $10^{-8} \leq \text{PFH} \leq 10^{-7}$ |
| SIL4 | e | $10^{-9} \leq \text{PFH} \leq 10^{-8}$ |

Tableau 3-8 : Le niveau de performance requis par valeur PFH [ISO 06]

Il faut noter que l'analyse des risques selon la norme EN ISO 13849 dépend non seulement du niveau de performance requis PLr, mais aussi des paramètres d'analyse des risques suivants :

- de la catégorie de sécurité du système qui, à son tour, découle de l'architecture du système et de sa résistance en cas de défaillance ;
- de la valeur *mean time to failure* (MTTF) des composants ;
- du facteur de couverture du diagnostic (DC) du système ;
- de la défaillance de cause commune CCF.

### 3.2.11 Facteur de couverture du diagnostic (*diagnostic coverage*)

Ce paramètre a pour but d'indiquer à partir de quel point le système est en mesure de s'autosurveiller face à une éventuelle défaillance dangereuse. En fonction du pourcentage de défaillances dangereuses détectables par le système, on aura une couverture du diagnostic plus ou moins élevée. Le paramètre numérique (DC) est une valeur processuelle issue des tableaux cités dans la norme ISO 13849-1, en fonction des précautions adoptées par le fabricant pour détecter les anomalies de son système.

Selon les genres de précaution à prendre, une valeur moyenne de couverture de diagnostic DC sera répartie dans les quatre groupes suivants :

| Classification | Valeurs DC |
|---|---|
| Nulle | 60 % < DC |
| Faible | 60 % < DC < 90 % |
| Moyenne | 90 % < DC < 99 % |
| Élevée | 99 % |

Tableau 3-9 : Les valeurs moyennes du facteur DC suivant ISO 13849 [ISO 06]

Le Tableau 3-10 [CEI 06] présente des prescriptions pour les anomalies qui doivent être détectées selon le type de composant par les techniques de maîtrise des défaillances du matériel afin d'obtenir ou bien de réaliser la couverture de diagnostic pertinente. Si, seules les erreurs de blocage des adresses ou bien des données sont détectées par un test de diagnostic en exploitation, cela signifie que la valeur du taux de couverture est faible de DC = 60 %, et si l'on ajoute des tests qui détectent les délais d'écoulement et le décodage d'adresse erronée, on atteint une couverture de diagnostic moyen de DC = 90 %.

| **Composant** | **Prescriptions pour la couverture de diagnostic** | | |
|---|---|---|---|
| | **Faible (60 %)** | **Moyen (90 %)** | **Élevé (99 %)** |
| Matériel discret E/S numérique | Blocage | Modèle CC | Modèle CC dérive et oscillation |
| Matériel discret E/S analogique | Blocage | Modèle CC dérive et oscillation | Modèle CC dérive et oscillation |
| Matériel discret Alimentation | Blocage | Modèle CC dérive et oscillation | Modèle CC dérive et oscillation |
| Mod**è**le CC (courant continu) indique des anomalies de blocage, blocage ouvert, sorties ouvertes ou haute impédance, ainsi que des courts-circuits entre les lignes des signaux. | | | |

Tableau 3-10 : Quelques anomalies et leur couverture de diagnostic [CEI 06]

### 3.2.12 Probabilité de défaillance dangereuse sur demande (PFD)

La probabilité de défaillance dangereuse sur demande (*probability of failure on demand*, PFD) est la probabilité sur l'intervalle de temps que le système ne puisse pas exécuter la fonction de sécurité pour laquelle il a été conçu au moment où la demande de cette fonction est faite. On obtiendra la probabilité de défaillance dangereuse à la demande par heure PFH en divisant la valeur PFD (Ti) par le temps de mission du système Ti.

La probabilité moyenne de défaillance à la demande PFDavg(t) est sur l'intervalle de temps [0, Ti] ; elle peut être obtenue à partir de la formule suivante [GBL 10] :

$$\mathrm{PFD}_{\mathrm{avg}}(\mathrm{T_i}) = \frac{1}{\mathrm{T_i}} \int_0^{\mathrm{T_i}} PFD(t)dt \qquad (3\text{-}4).$$

### 3.2.13 Système par niveau de sécurité (*safety integrity level SIL*)

Le niveau de sécurité SIL se trouve dans presque toutes les normes qui sont élaborées à partir de la norme CEI 61508, à savoir la norme EN 61511 pour les procédés industriels, la norme EN 61513 pour le nucléaire, les normes EN 50128/50129 pour le secteur ferroviaire, ou encore la norme EN 62061 pour le secteur automobile.

La norme de sécurité CEI 61508 [CEI 06] vise à quantifier la probabilité de défaillance, ce qui permet de classer les systèmes par niveau de sécurité SIL. Le SIL définissant donc la probabilité de défaillance dangereuse que l'on s'autorise. Selon [CEI 06], le SIL ne peut prendre que 4 valeurs possibles (de 1 à 4). Il est donc inutile de chercher à définir les valeurs précises des probabilités de défaillance dangereuse, la valeur obtenue doit être comprise dans l'intervalle défini par le SIL. Ces quatre niveaux sont décrits dans le Tableau 3-11. Le Tableau 3-11 donne les correspondances entre les probabilités de défaillance, les niveaux de sécurité et le niveau de performance PL.

| **Niveaux de sécurité** | **Niveau de performance PL** | **Probabilité de défaillance dangereuse par heure** | **Probabilité de défaillance sur demande** |
|---|---|---|---|
| SIL4 | e | $10^{-9} \leq \mathrm{PFH} \leq 10^{-8}$ | $10^{-5} \leq \mathrm{PFD} \leq 10^{-4}$ |
| SIL3 | d | $10^{-8} \leq \mathrm{PFH} \leq 10^{-7}$ | $10^{-4} \leq \mathrm{PFD} \leq 10^{-3}$ |
| SIL2 | c | $10^{-7} \leq \mathrm{PFH} \leq 10^{-6}$ | $10^{-3} \leq \mathrm{PFD} \leq 10^{-2}$ |
| SIL1 | b | $10^{-6} \leq \mathrm{PFH} \leq 10^{-5}$ | $10^{-2} \leq \mathrm{PFD} \leq 10^{-1}$ |

Tableau 3-11 : Intégrité de sécurité du matériel [CEI 06]

### 3.2.14 Méthodes de modélisation

La valeur de la probabilité de défaillance dangereuse d'un système peut être calculée à partir des trois types de modélisations suivantes [GBL 10] :

- Modélisation par des blocs-diagrammes de fiabilité.
- Modélisation à partir de l'arbre des causes. Cette méthode de modélisation du système consiste à modéliser le comportement du système en présence des fautes possibles.
- Modélisation grâce à des graphes de Markov, qui décrivent les états intermédiaires du système.

### 3.2.14.1 Méthode du diagramme de fiabilité

Le diagramme de fiabilité consiste à construire un diagramme composé de blocs, chacun d'eux représente une entité (composant, sous-système) et reliés par des arcs indiquant les dépendances des entités entre elles. Le but de cette représentation est d'obtenir une modélisation statique du fonctionnement du système, ce qui consiste à chercher les combinaisons de défaillances d'entités élémentaires conduisant à la défaillance totale du système. La transmission d'un signal entre les entités peut se produire en série ou bien en parallèle.

Une défaillance au niveau d'une entité dans un système en série entraînera l'arrêt du signal au niveau du bloc (Figure 3-9) qui lui est associé, impliquant l'arrêt du système.

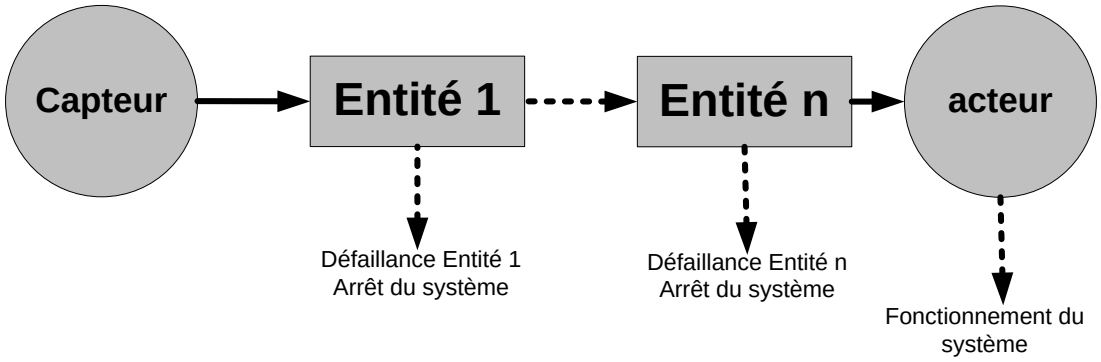



Figure 3-9 : Le diagramme de fiabilité en série

À l'inverse, dans un système en parallèle, le fonctionnement d'une seule entité suffit au passage du signal. L'arrêt du système n'est possible que si les trois entités sont défaillantes.

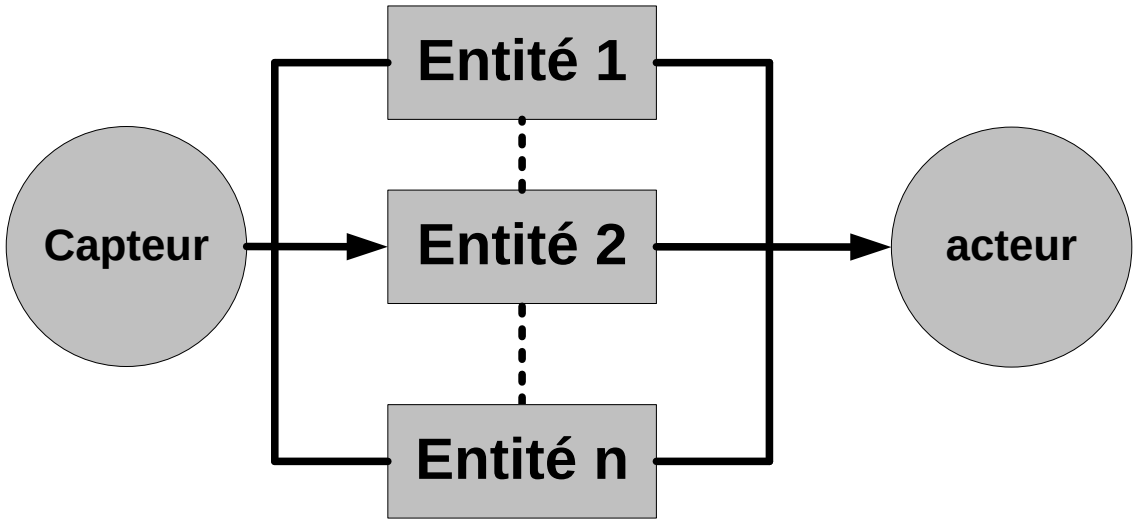



Figure 3-10 : Le diagramme de fiabilité en parallèle

### 3.2.14.2 Méthode d'arbre des causes

L'arbre des causes est une représentation graphique des relations logiques entre les défaillances et les événements qui peuvent leur être associés, qui se traduisent, en service, par une logique binaire de « 1 », ou bien hors service, par une logique binaire « 0 ». La connaissance exacte des causes de tous les événements de défaillances possibles ne donne pas la possibilité de prendre des décisions effectives sur le système. On peut notamment décider d'exclure certaines défaillances ou bien de réduire leurs impacts négatifs, ce qui augmente la sécurité du système. L'arbre des causes est constitué de plusieurs niveaux d'événements qui sont reliés entre eux de façon à ce que chaque événement soit directement lié à un certain niveau. Les liaisons des événements sont réalisées par différents opérateurs logiques. On peut compter par exemple parmi les événements les défaillances de matériel, de l'opérateur, ou encore les défaillances du logiciel qui peuvent avoir des conséquences néfastes.

### 3.2.14.3 Méthode chaînes de Markov

La méthode des chaînes de Markov [GHY 08], [MSB 10] apporte une bonne formalisation des états que peuvent prendre un système instrumenté de sécurité (SIS) en fonction des événements rencontrés et des paramètres étudiés (taux de défaillances, facteur de DCC).

Avec l'aide des chaînes de Markov, un système peut être analysé sur une longue période de façon à prédire le processus futur du système en se basant sur l'état présent du processus. La loi de transition d'une chaîne de Markov est définie par l'équation suivante [MSB 11] et [GBL 10] :

$$p^{n}(S_j) = \sum_{i} p^{(n-1)}(S_j) \cdot \lambda_{ij} \qquad (3\text{-}5)$$

$p^{n}(S_j)$ est la mesure de la probabilité d'être dans l'état Sj à l'instant n et les λij représentent les taux de transition de l'état Si vers l'état Sj.

L'équation représente la probabilité que le système étudié soit dans l'état Sj à l'instant n à partir de n'importe quel autre état Si à l'instant (n-1) selon une probabilité de transition λij de Si vers Sj définie dans la matrice de transition M = (λij). La probabilité moyenne de défaillance à la demande est calculée comme suit [MSB 11] et [GBL 10] :

$$PFDavg = \frac{1}{k \cdot \Delta t} \cdot \sum_{n=0}^{k} \sum_{Sj} p^{(n)}(S_j) \cdot \lambda_{ij} \cdot \Delta t \quad \text{avec } k \cdot \Delta t \in [0, T] \qquad (3\text{-}6)$$

Ti est le temps de mission, Si sont les états de défaillances dangereuses et est la probabilité d'être dans un de ces états à l'instant n.

### 3.2.15 Analyse de mode de défaillance et effets (AMDEC)

L'AMDEC est équivalent au mode de défaillance et analyse des effets. L'idée à l'origine de la méthode AMDEC est d'essayer de prévenir les causes possibles de défaillance du système pour éliminer des pannes qui peuvent se produire pendant la phase de développement jusqu'à la phase de livraison du système. Il est donc impératif de s'assurer que les risques et les vulnérabilités du système soient détectés le plus tôt possible pour effectuer des améliorations. Tous les éléments critiques devront être filtrés, et les modes de défaillances possibles du système devront être déterminés. Par la suite, ces modes de défaillances sont analysés en fonction de leur impact sur la fonctionnalité et la sécurité. Ces évaluations sont incluses dans une prétendue check-list. Cette technique est principalement utilisée dans les domaines de l'aérospatial, de l'ingénierie nucléaire et de l'ingénierie automobile.

La force de cette méthode réside dans l'exhaustivité de l'analyse. Malheureusement, sa mise en œuvre dépend également de questions de coût et de temps.

## 3.3 Évaluations qualitative et quantitative selon la norme de sécurité 61508

La norme de sécurité CEI 61508 est une norme internationale pour la sécurité fonctionnelle (*functional safety*) des équipements électriques, électroniques et électroniques programmables. Cette norme est apparue dans le milieu des années quatre- vingt lorsque le comité consultatif Commission internationale électrotechnique de sécurité (*International electrotechnical committee advisory committee of safety*, IEC ACOS) a mis en place un groupe de travail pour examiner les questions soulevées par la normalisation de l'utilisation des systèmes électroniques programmables (*programmable electronic systems*, PES). À cette époque, de nombreux organismes de réglementation avaient interdit l'utilisation de tous équipements électriques et électroniques programmables dans des applications critiques de sécurité.

La norme CEI 61508 [CEI 06] est divisée en sept parties :

- partie 1 : Prescriptions générales ;
- partie 2 : Prescriptions pour les systèmes électriques et électroniques ou électroniques programmables relatifs à la sécurité ;
- partie 3 : Prescriptions concernant les logiciels ;
- partie 4 : Définitions et abréviations ;
- partie 5 : Exemples de méthodes pour la détermination des niveaux d'intégrité de sécurité ;
- partie 6 : Lignes directrices concernant l'application des parties 2 et 3 ;
- partie 7 : Présentation de techniques et mesures.

Les parties 1, 3, 4 et 5 ont été approuvées en 1998. Les parties 2, 6 et 7 ont été approuvées en février 2000.

Bien que la norme ait été critiquée pour sa documentation extensive, elle a permis de nombreuses avancées dans beaucoup d'industries.

La norme met l'accent sur la conception du système fondé sur les risques liés à la sécurité, rendant le système beaucoup plus rentable.

En raison de ces caractéristiques et du degré élevé d'acceptation internationale pour un ensemble unique de documents, beaucoup la considèrent comme une avancée majeure pour le monde de la technique.

Pour l'évaluation de la fonction de sécurité dans notre moteur d'inférence floue, le processus de la conception défini dans la norme CEI 61508 sera utilisé. Le logigramme suivant représente les différentes étapes de conception d'une fonction de sécurité :

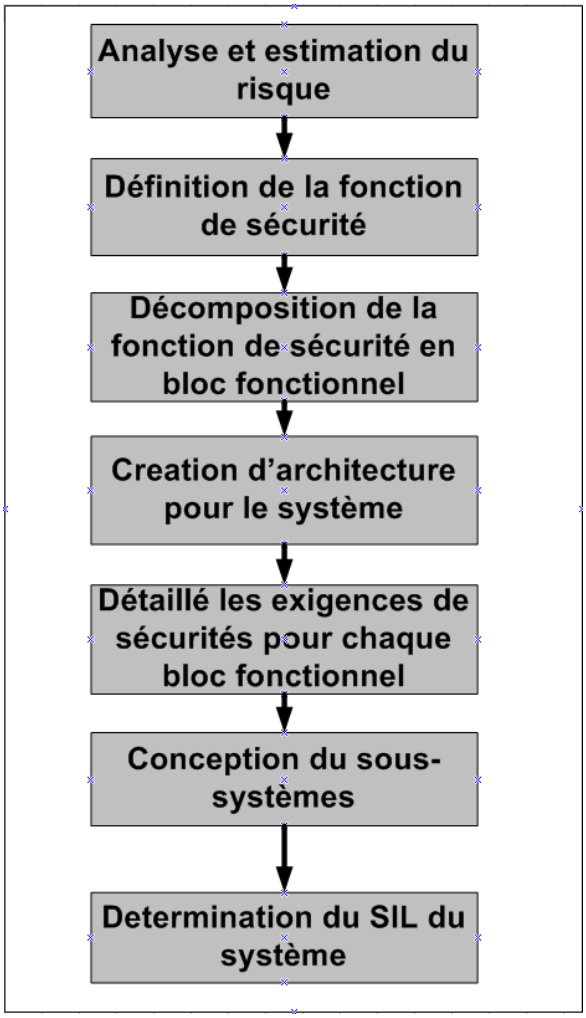



Figure 3-11 : Conception d'une fonction sécurité

## 3.4 Conclusion

On a présenté les différentes terminologies nécessaires à la compréhension d'un système instrumenté de sécurité (SIS) : sécurité fonctionnelle, fiabilité, disponibilité, maintenabilité, sûreté fonctionnement, taux de défaillance, niveau de performance requis, probabilité de défaillance dangereuse et le niveau de sécurité. On a présenté également la structure des différentes architectures dédiées à ces systèmes à savoir les architectures un parmi un (*1 out of 1*), un parmi un avec diagnostic un (*1 out of 1D*), un parmi deux (*1 out of 2*), un parmi deux avec diagnostic (*1 out of 2D*), deux parmi deux (*2 out of 2*), ainsi qu'une analyse détaillée des pannes qui peuvent affecter le MIF avec les tests de diagnostic et les tests d'inspection associés. On présente également la modélisation de calcul de la valeur de la probabilité de défaillance dangereuse d'un système avec différentes architectures en utilisant trois méthodes : modélisation par des blocs-diagrammes de fiabilité, modélisation à partir de l'arbre des causes, méthode qui consiste à modéliser le comportement du système en présence des fautes possibles et enfin, modélisation grâce à des graphes de Markov, qui décrivent les états intermédiaires du système.

# Chapitre 4

# Évaluation qualitative du moteur d’inférence floue

## *Résumé du chapitre 4 :*

*Ce chapitre est consacré à l'évaluation qualitative du régulateur flou avec la représentation de l'architecture redondante homogène du système. Le chapitre présente l'analyse des différentes architectures dédiées aux applications de la sécurité (1oo1, 1oo1D ,1oo2, 1oo2D, 2oo2,) du moteur d'inférence floue à partir de la détermination de la probabilité moyenne de défaillance sur demande PFDavg, du temps de l'indisponibilité et le facteur de réduction de risque pour chaque structure. Ceci afin de pouvoir choisir la structure qui répond le mieux aux exigences de la sécurité. Ainsi, et en partant du graphe de risque, une estimation du risque pour maîtriser le risque sera effectuée.*

# 4 Évaluation qualitative du moteur d'inférence floue

## 4.1 Introduction

À l'heure actuelle, le marché des composants électroniques basés sur la technologie FPGA est de plus en plus complexe. En effet, ces systèmes peuvent contenir au sein d'une même puce jusqu'à quatre processeurs exigeant des méthodes de conception adaptées à des normes de sécurité afin d'éviter les erreurs qui sont susceptibles d'amener le système dans un état dangereux causant des incidents très graves. Une analyse du risque et un flot de conception qui répond aux contraintes de la sûreté de fonctionnement doivent être effectués.

Dans ce chapitre, on va analyser les différentes architectures dédiées aux applications de la sécurité (1oo1, 1oo1D, 1oo2, 1oo2D, 2oo2) sur notre moteur d'inférence floue à partir de la détermination de la probabilité moyenne de défaillance sur demande PFDavg, du temps de l'indisponibilité et le facteur de réduction de risque pour chaque structure. Afin de pouvoir choisir la structure qui répond aux exigences de la sécurité.

À partir du graphe de risque (Figure 4-1), on effectuera tout d'abord une estimation du risque qu'ainsi une allocation du niveau de performance pour maîtriser le risque. Si le moteur d'inférence floue (MIF) subit une défaillance dangereuse, cela introduira des pertes du matériel et du personnel, on peut donc quantifier cela par une gravité de blessure grave (S = S2) qui peut mener dans le pire de cas à un décès. Le moteur d'inférence floue est en service 24 heures sur 24 heures alors que la fréquence d'exposition à un phénomène dangereux est de longue durée (F = F2). Le flot de conception décrit dans cette thèse rend possible d'éviter ces défaillances dangereuses jusqu'à les limiter (P = P1).

À l'aide de ce graphe du risque, les fonctions de sécurité à implanter pour prévenir un danger de forte probabilité seront réalisées en tenant compte des exigences relatives au niveau de performance requis PLd.

Comme nous l'avons vu dans le chapitre précédent, l'évaluation qualitative du MIF avec sécurité dépend non seulement du niveau de performance, mais aussi de plusieurs paramètres, à savoir :

1) De la structure du système du MIFS et de sa résistance en cas de défaillance.

2) De la probabilité de défaillance dangereuse PFDAVG des composants du MIFS.

3) Des facteurs de taux de couverture, qui sont définis à partir du genre de tests de diagnostic présents.

4) Des défaillances de cause commune CCF.

5) Des tests de diagnostics.

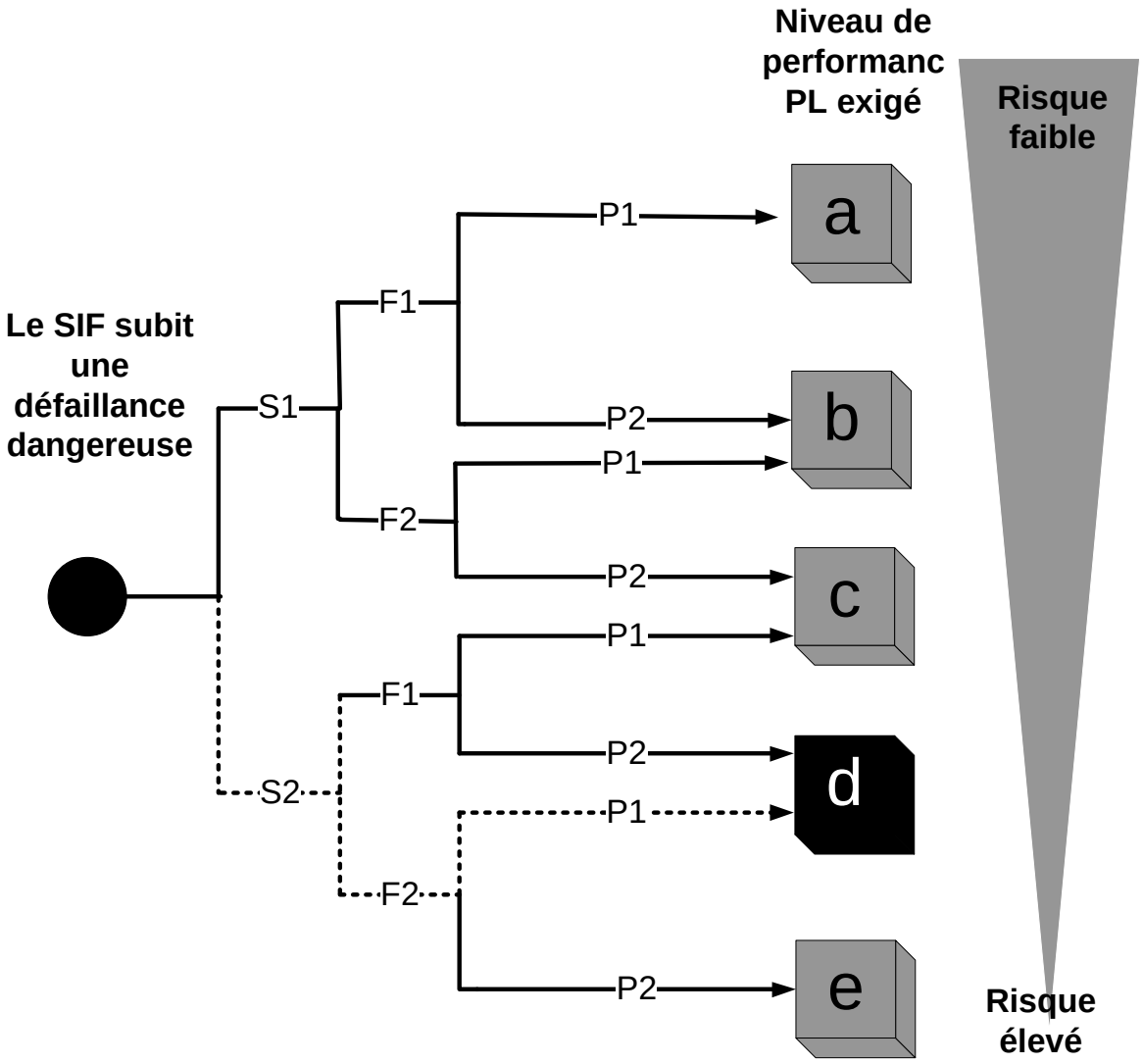



Figure 4-1 : Niveau de performance requis PLd du MIF

En notant que les tests de diagnostics qui seront présents dans le MIFS font partie de l'analyse des différentes architectures. Deux situations peuvent être générées si une défaillance est détectée :

- Soit la défaillance est signalée pour être réparée dans un délai de réparation Tr égal à 8 heures en notant que, durant ce temps mort entre l'apparition de cette défaillance et sa réparation, une autre défaillance dangereuse peut survenir dans le cas d'une architecture simple d'un seul canal (architecture 1oo1), menant le MIFS à un état de défaillance dangereuse. Dans ce cas, avec une structure redondante, le MIF peut effectuer la fonction de sécurité jusqu'à la réparation du MIFS défaillant (structure 1oo2).
- Soit la défaillance provoque automatiquement le passage à un état de défaillance sûre, ou dans le mode dégradé, à un état de défaillance détectée, dans le but de continuer à assurer la fonction de sécurité jusqu'à la réparation.

Une défaillance dangereuse détectée n'est donc pas si « dangereuse » (1oo2D).

Dans la suite, les différentes architectures de notre moteur d'inférence floue seront représentées et analysées en tenant compte des paramètres mentionnés ci-dessus.

## 4.2 Moteur d'inférence floue traditionnel

Le moteur d'inférence floue traditionnel (Figure 4-2) se compose d'un convertisseur analogique, d'un contrôleur flou qui est constitué d'un processus de fuzzification, d'un processus pour l'établissement des règles, et d'un processus de défuzzification :

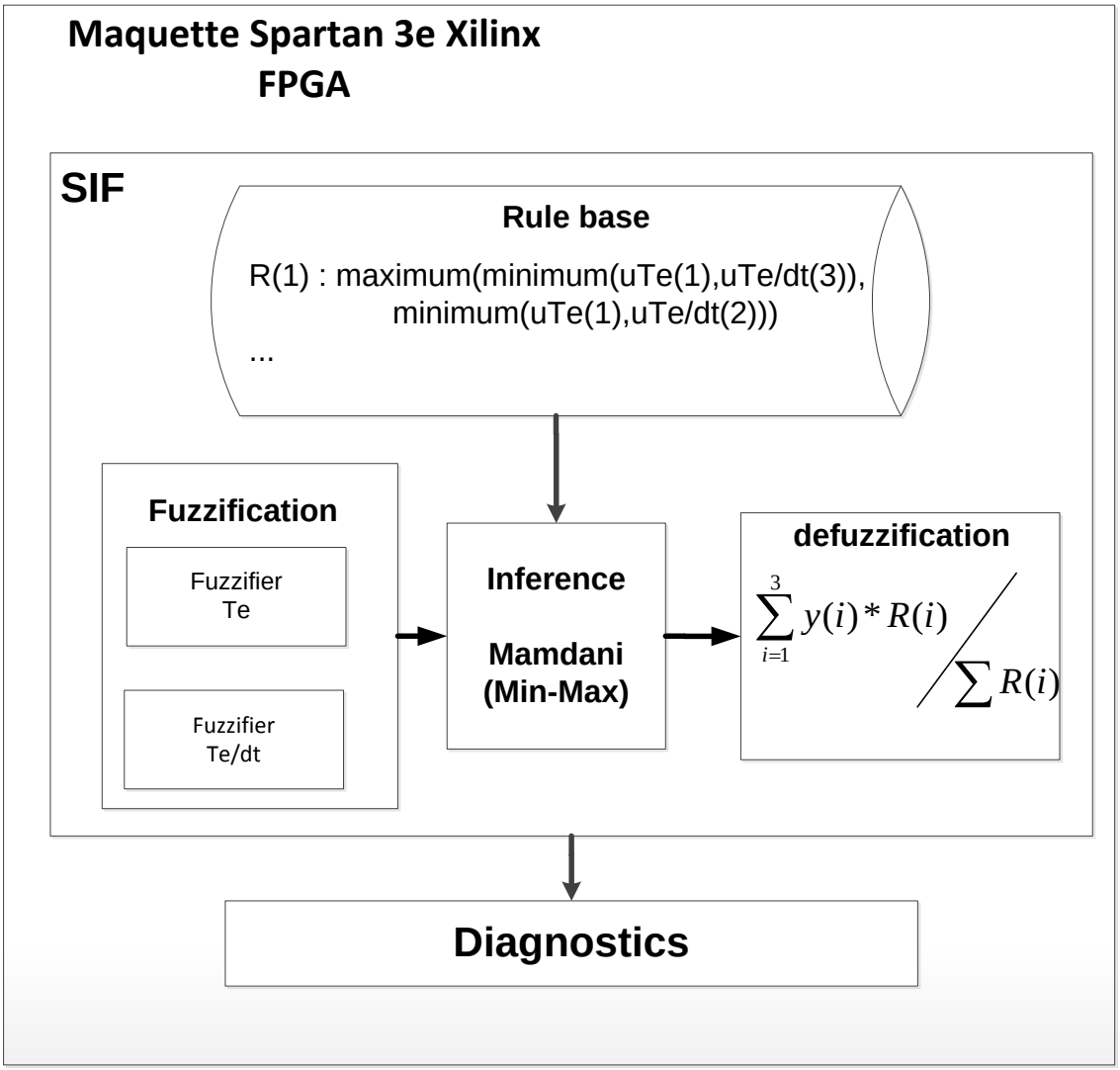



Figure 4-2 : Schéma de principe d'une architecture en général

Elle peut trouver son application par exemple dans le contrôle de la température des chambres de refroidissement par la commande du moteur de l'aération. La Figure 4-3 représente le modèle de base avec capteur et actionneur.

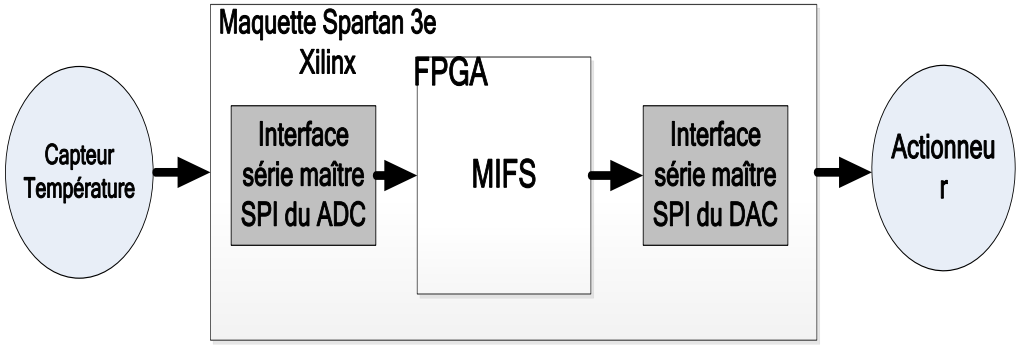



Figure 4-3 : Moteur d'inférence floue

Chaque élément de la chaîne de commande (capteur, commande, actionneur) peut subir une défaillance dangereuse qui peut mener à des incidents très graves. Une analyse du risque doit être effectuée pour classer les différentes erreurs éventuelles et les précautions qu'on doit prendre pour prévenir ou atténuer l'événement dangereux pour chaque élément de la chaîne.

Dans notre étude, on ne traitera que la partie commande, plus précisément on traitera la mise en œuvre d'un moteur d'inférence floue à base de FPGA, pour atteindre un niveau de sécurité qui sera fiable. À partir du diagramme représenté par la Figure 4-4, la classification des défaillances du MIF en défaillances dangereuses et sûres sera effectuée.

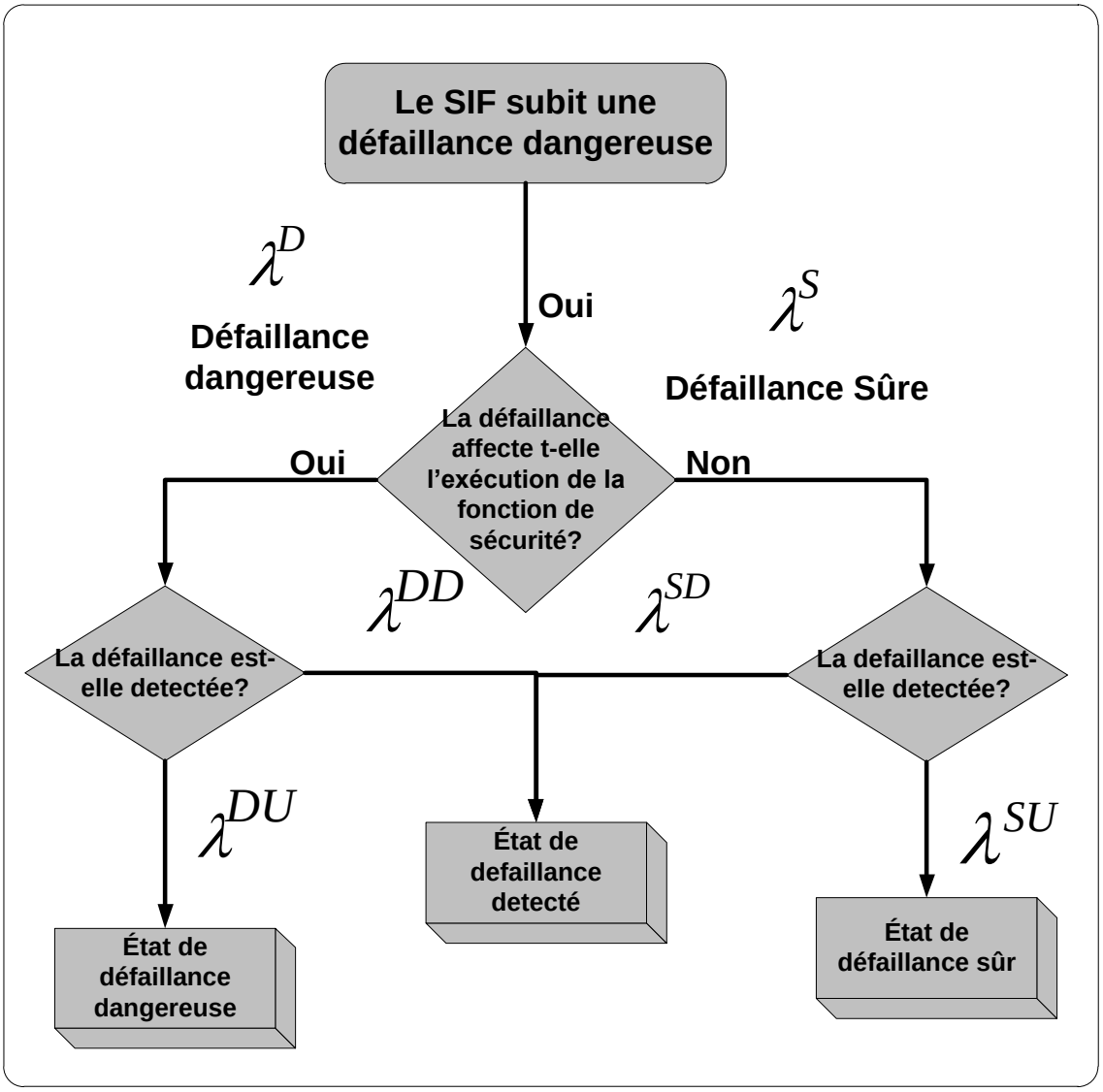



Figure 4-4 : Diagramme d'état de défaillance

Les défaillances possibles du moteur d'inférence floue implémenté en FPGA et leur classification sont présentées dans le Tableau 4-1 et le Tableau 4-2 :

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **Genre de défaillance** | **Causes potentielles** | **Test de diagnostic** | **Classification de la défaillance** |
| Défaillance du processus de fuzzification. | Anomalie de blocage au niveau des éléments internes FPGA utilisés pour ce processus. | Comparaison périodique du résultat des deux contrôleurs flous. | Défaillance dangereuse détectée. |
| Défaillance du processus d'établissement des règles. | | | |
| Défaillance du processus de défuzzification dans le circuit FPGA. | | | |
| Défaillance d'un élément interne qui n'intervient pas dans la logique implémentée dans FPGA. | Anomalie de blocage au niveau des éléments internes FPGA utilisés. | Pas de diagnostic. | Comme elle n'affecte pas la fonction de sécurité du MIF alors il s'agit d'une défaillance sûre non détectée. |
| Défaillance de la mémoire flash où la logique (code VHDL) sera mémorisée. | Défaut du matériel, perturbation électrostatique, les ondes magnétiques, les fréquences de haute tension, etc. | Examen de l'intégrité du logiciel du MIF mis dans la mémoire flash par une valeur CRC. | Une défaillance au niveau de la mémoire du flash pendant le fonctionnement du MIF peut être détectée seulement après le délai du temps de mission Ti. On peut donc la classifier comme une défaillance sûre détectée. |
| La dérive de l'horloge principale (master). | Défaut du matériel, perturbation électrostatique, les ondes magnétiques, les fréquences de haute tension, etc. | Le chien de garde. | Défaillance dangereuse détectée. |

Tableau 4-1 : L'analyse des défaillances et leur classification du MIF

| La fonction de sécurité est assurée si la sortie n'est plus alimentée | | | |
|---|---|---|---|
| **Genre de défaillance** | **Causes potentielles** | **Test de diagnostic** | **Classification de la défaillance** |
| Chute ou bien augmentation rapide de la température. | Le capteur est défaillant, chute ou bien augmentation rapide de la température dans la chambre de refroidissement. | Contrôle des gammes de valeurs limites de la consigne (température). | Défaillance dangereuse détectée. |
| Détection du saut de ligne au niveau du capteur. | Si le capteur subit une rupture au niveau des fils d'entrée alors on parle de saut de la valeur analogique d'entrée et par conséquent de la valeur de la température. | Pas possible dans cette carte | Défaillance dangereuse non détectée.<br>Pas possible à détecter par manque du moyen sur la carte. |
| Court-circuit au niveau d'alimentation. | Perturbation par onde électrostatique ou bien magnétique. Une fausse utilisation de la charge. | Cela n'a pas de sens de réaliser un test de diagnostic, car la carte dans ce cas ne fonctionnera pas. On a besoin d'une alimentation redondante pour assurer le bon fonctionnement du MIF. | Défaillance dangereuse non détectée.<br>Pas possible de détecter par manque du moyen sur la carte. |

Tableau 4-2 : L'analyse des défaillances et leur classification du MIF (suite)

## 4.3 Moteur d'inférence floue d'une architecture simple

Le moteur d'inférence floue d'une structure simple un parmi un (*1 out of 1*) est comme le MIF traditionnel. Il est constitué d'un processus de fuzzification, d'un processus pour l'établissement des règles, et d'un processus de défuzzification, formant ainsi une structure à un seul canal pour laquelle une défaillance dangereuse au niveau du canal entraînera le contrôleur flou dans un état dangereux. Les tests de diagnostic pour cette structure sont présents uniquement dans le but de réparer le système après une défaillance détectée.

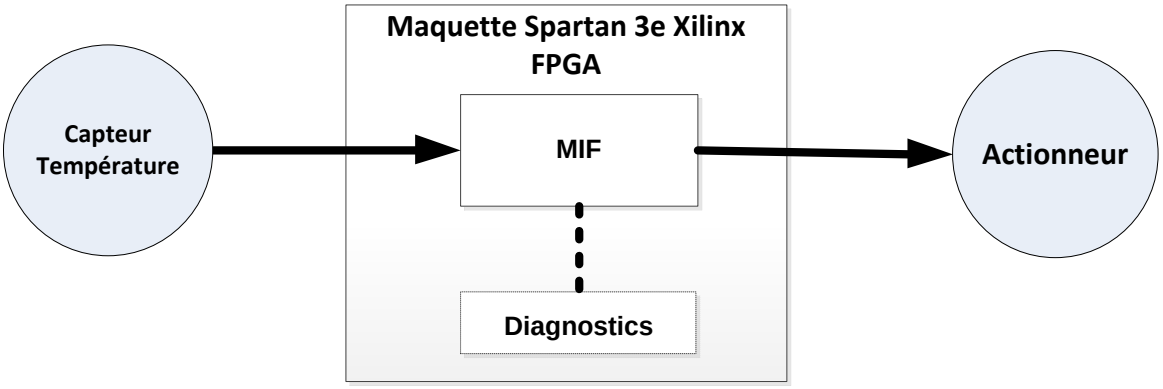



Figure 4-5 : Schéma du principe pour une structure 1oo1

Le Tableau 4-3 montre les différents états du système un parmi un (1oo1). Par définition, la fonction de sécurité se résume à une rupture de la tension de sortie en cas d'une détection d'une défaillance dangereuse au niveau du canal.

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **État du système** | **Cause potentielle** | **Disponibilité du système** | **Sécurité du système** |
| Canal défaillant | Une défaillance dangereuse (voir le Tableau 4-1 et le Tableau 4-2). | Impossibilité d'actionner la sortie. | Le système n'est plus sûr. |

Tableau 4-3 : La disponibilité et la sécurité dans une architecture 1oo1

Note système possède un canal qui peut être défaillant avec un taux de défaillances dangereuses $\lambda^D$ résultant des défaillances non détectées $\lambda^{DU}$ ou bien avec un taux de défaillances dangereuses $\lambda^{DD}$ résultant des défaillances détectées. Le temps d'indisponibilité $t_{CE}$ du canal est calculé par l'addition du temps de l'indisponibilité du canal à cause d'une défaillance dangereuse non détectée $t_{C1}$ et le temps de l'indisponibilité du canal à cause d'une défaillance dangereuse détectée $t_{C2}$ comme suit [CEI 06] :

$$t_{C1} = \frac{T_i}{2} + MTTR \qquad (4\text{-}1)$$

$$t_{C2} = MTTR \tag{4-2}$$

$$t_{CE} = \left( \frac{\lambda^{DU}}{\lambda^{D}} (\frac{T_i}{2} + MTTR) + \frac{\lambda^{DD}}{\lambda^{D}} MTTR \right) \tag{4-3}$$

Avec :

- MTTR (*mean time to repair*), est le temps de réparation conventionnelle, il est de 8 heures ; c'est-à-dire après la détection de la première défaillance dangereuse, le responsable dispose d'une durée de 8 heures pour résoudre le problème (remplacement de la carte si une redondance existe), sinon après ce temps le système n'est plus fiable.
- Ti est le temps de mission (on utilise souvent 1 an, 3 ans, 5 ans ou bien 10 ans). C'est-à-dire après ce temps, on doit effectuer un redémarrage du système pour contrôler le système par les tests d'inspection.
- $\lambda^{DD}$ Taux de défaillance dangereuse détectée.
- $\lambda^{DU}$ Taux de défaillance dangereuse non détectée.

La Figure 4-6 montre le schéma de principe de la fiabilité pour une architecture de 1oo1.

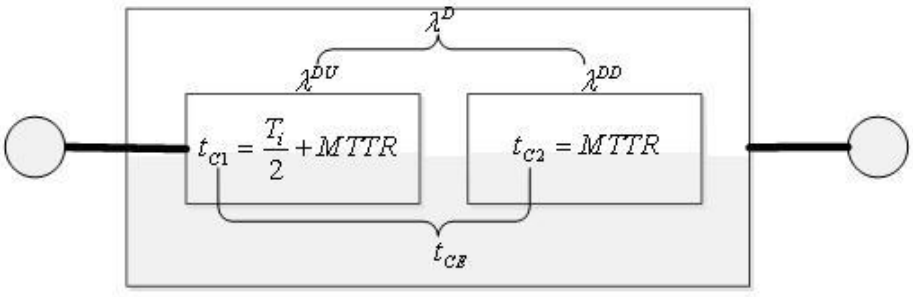



Figure 4-6 : Schéma de principe de la fiabilité pour une structure 1oo1

À partir de la fiabilité R(t) du système on déduit, la probabilité d'une défaillance dangereuse comme suit [CEI 06] :

$$PFD = 1 - e^{-t_{CE}\lambda^{D}} = t_{CE}\lambda^{D} \text{ puisque } t_{CE}\lambda^{D} << 1 \tag{4-4}$$

Ainsi, la probabilité moyenne de défaillance sur demande est [CEI 06] :

$$PFDavg = \lambda^D t_{CE} \tag{4-5}$$

Le facteur de réduction du risque RRF est calculé à partir de la formule suivante [GBL 11] :

$$RFF = \frac{1}{PFD} \tag{4-6}$$

Le taux de défaillance dangereuse non détecté du moteur d'inférence floue basé sur la technologie FPGA de la famille Spartan-3E égale à la somme du taux de défaillance de la couche de configuration représentée par 104 FIT et de la couche opérative représentée par la valeur 293 FIT (Tableau 4-4), ce qui donne un taux de défaillance 397 FIT.

| **Technologie node** | **La famille FPGA** | **La valeur FIT la couche configuration** | **La valeur FIT la couche opérative** |
|---|---|---|---|
| 90 nm | Spartan-3E | 104 | 293 |
| 28 nm, | FPGA 6 | 86 | 78 |

Tableau 4-4 : La technologie FPGA et le taux de défaillance [DRR 00]

Étant donné le taux de défaillance de base pour chaque famille FPGA, le taux de défaillance dangereuse détectée et le taux de défaillance dangereuse non détectée peuvent être déduits par les formules suivantes :

$$\lambda^{DU} = \lambda^D(1-DC) \quad , \quad \lambda^{DD} = \lambda^D DC \quad et \quad \lambda^D = \frac{\lambda}{2} \tag{4-7}$$

Avec DC le facteur de couverture du diagnostic.

L'influence du test de diagnostic sur les valeurs des taux de défaillances du MIF de structure 1oo1 pour la puce FPGA de la famille Spartan-3E et Spartan-6 pour un temps de mission Ti = 1 [an] est représenté par le Tableau 4-5 .

Le calcul de taux de défaillances dangereuses (Tableau 4-5) montre qu'un moteur d'inférence floue de structure 1oo1 implémenté sur une puce FPGA de la famille Spartan-6 sera deux fois plus résistant à des défaillances dangereuses qu'une implémentation de ce dernier dans une puce FPGA de la famille Spartan-3E

| Famille FPGA | DC | Taux de défaillance dangereuse détectée [$h^{-1}$] | Taux de défaillance dangereuse non détectée [$h^{-1}$] |
|---|---|---|---|
| Spartan-3E | Taux de défaillance Spartan 3-E, λ=397 E-09 par [h] | | |
| | 0 % | 0,00 E+00 | 1,99 E-07 |
| | 60 % | 1,19 E-07 | 7,94 E-08 |
| | 90 % | 1,79 E-07 | 1,99 E-08 |
| | 99 % | 1,97 E-07 | 1,99 E-09 |
| Spartan-6 | Taux de défaillance Spartan-6, λ =164 E-09 par [h] | | |
| | 0 % | 0,00 E+00 | 8,20 E-08 |
| | 60 % | 4,92 E-08 | 3,28 E-08 |
| | 90 % | 7,38 E-08 | 8,20 E-09 |
| | 99 % | 8,12 E-08 | 8,20 E-10 |

Tableau 4-5 : L'influence du facteur DC sur le taux de défaillance

.On constate aussi (Tableau 4-5) que l'impact du taux de la couverture de diagnostic (DC) sur les valeurs des taux de défaillance sera grand. On constate qu'un moteur flou MIF avec des tests de diagnostic d'une valeur faible de 60 % pourra être deux fois plus défaillant qu'un MIF avec un DC de 99 %. D'autre part le temps d'indisponibilité du MIF sera réduit de 1 760 heures pour une couverture de diagnostic DC = 60 % à 52 heures pour une couverture de diagnostic DC = 99 %.

La probabilité moyenne de défaillance sur demande de la fonction de sécurité pour les deux types de circuits PGA pour un temps de mission Ti = 1 [an] est représentée par le Tableau 4-6 :

| | | La valeur PFDavg par [an] | |
|---|---|---|---|
| Architecture | DC | Taux de défaillance Spartan-3E λ=397 E-09 par [h] | Taux de défaillance Spartan-6 λ =164 E-09 par [h] |
| MIF de structure1oo1 | 0 % | 8,71 E-04 | 3,60 E-04 |
| | 60 % | 3,49 E-04 | 1,44 E-04 |
| | 90 % | 8,85 E-05 | 3,66 E-05 |
| | 99 % | 1,03 E-05 | 4,25 E-06 |

Tableau 4-6 : Le choix du composant et la valeur DC

On constate ainsi Tableau 4-6 l'influence du choix des composants et les tests de diagnostic (DC) sur la valeur de la probabilité moyenne de défaillance sur demande.

La probabilité moyenne de défaillance sur demande du MIF mise en œuvre dans une puce FPGA de la maille Spartan-3E sera deux fois plus élevée qu'une implémentation du MIF sur une puce FPGA de la famille 6.

On constate aussi que l'influence des tests de diagnostic sur la valeur PFDavg est majeure. La probabilité moyenne de défaillance sur demande des deux composants FPGA avec une couverture de diagnostic DC = 60 % sera 100 fois plus élevée par rapport à une implémentation du MIF avec une couverture de diagnostic de DC = 99 %.

L'influence du temps de mission sur la probabilité de défaillance dangereuse est présentée sur le Tableau 4-7.

L'impact de temps de mission sur la valeur PFDavg nous permet de prévoir le comportement du système vis-à-vis des défaillances dangereuses. On constate d'après le Tableau 4-7 que si l'on retarde les tests d'inspection (le redémarrage du système) d'une période de 10 ans, la probabilité que notre système subisse une défaillance dangereuse sera augmentée de 10 fois par rapport à un redémarrage du système après chaque année de service.

Comme la famille de Spartan-3E répond aux contraintes de niveau de sécurité exigées de PLd, le calcul pour les autres structures sera fait seulement pour cette famille.

| | | Probabilité moyenne de défaillance sur demande PFDavg | | | |
|---|---|---|---|---|---|
| **La famille FPGA** | **DC** | **Ti = 1 an** | **Ti = 3 ans** | **Ti = 5 ans** | **Ti = 10 ans** |
| **Spartan-3E** | **0 %** | 8,71 E-04 | 2,61 E-03 | 4,35 E-03 | 8,70 E-03 |
| | **60 %** | 3,49 E-04 | 1,04 E-03 | 1,74 E-03 | 3,48 E-03 |
| | **90 %** | 8,85 E-05 | 2,62 E-04 | 4,36 E-04 | 8,71 E-04 |
| | **99 %** | 1,03 E-05 | 2,77 E-05 | 4,51 E-05 | 8,85 E-05 |
| **Spartan-6** | **0 %** | 3,60 E-04 | 1,08 E-03 | 1,80 E-03 | 3,59 E-03 |
| | **60 %** | 1,44 E-04 | 4,32 E-04 | 7,19 E-04 | 1,44 E-03 |
| | **90 %** | 3,66 E-05 | 1,08 E-04 | 1,80 E-04 | 3,60 E-04 |
| | **99 %** | 4,25 E-06 | 1,14 E-05 | 1,86 E-05 | 3,66 E-05 |

Tableau 4-7 : L'impact du Ti sur la valeur PFDavg sur le MIF 1d'une structure 1oo1

## 4.4 Moteur d'inférence floue d'une architecture simple avec diagnostic

Le moteur d'inférence floue avec une architecture un parmi un avec diagnostic (1oo1D) possède en plus de celui d'une structure simple, des tests de diagnostics qui sont capables d'agir sur la sortie en cas d'une détection d'une défaillance dangereuse. La Figure 4-7 représente la structure du MIF de structure 1oo1D. Le système passe dans un état sûr, si les tests de diagnostic détectent au moins une défaillance dangereuse.

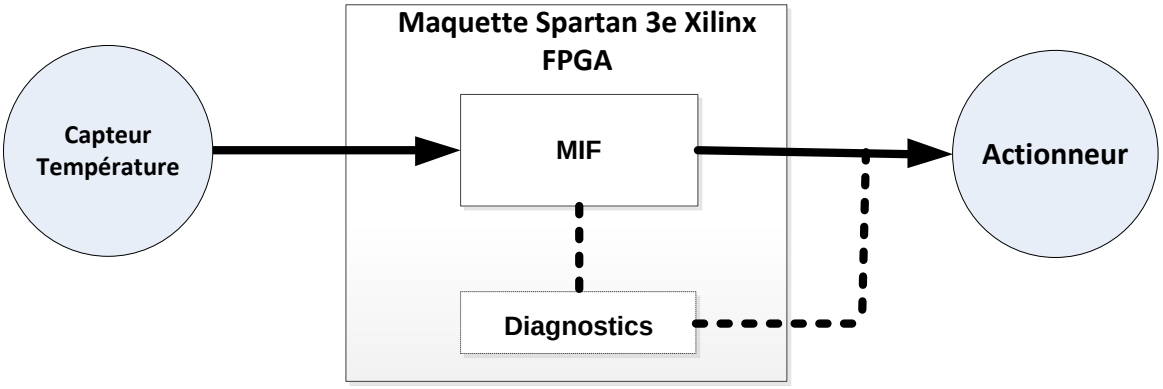



Figure 4-7 : Schéma du principe pour une structure 1oo1D

Le Tableau 4-8 montre les différents états du système 1oo1D.

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **État du système** | **Causes potentielles** | **Disponibilité du système** | **Sécurité du système** |
| Le canal est défaillant | Une défaillance dangereuse (voir le Tableau 4-1 et le Tableau 4-2). | Défaillance détectée<br>Possibilité d'actionner la sortie par les tests de diagnostic. | Le système est sûr. |
| | | Défaillance non détectée<br>Impossibilité d'actionner la sortie par les tests de diagnostic. | Le système n'est plus sûr. |

Tableau 4-8 : La disponibilité et la sécurité dans une architecture 1oo1D

Selon [CEI 06] le temps d'indisponibilité $t_{CE}'$ du canal est calculé par l'addition du temps $t_{C1}$ de l'indisponibilité du canal à cause d'une défaillance dangereuse non détectée et le temps $t_{C2}$ de l'indisponibilité du canal à cause d'une défaillance dangereuse détectée ainsi que le temps $t_{C3}$ , ( $t_{C3} = t_{C2}$ ) de l'indisponibilité du canal à cause d'une défaillance sûre détectée [CEI 06] :

$$t_{C1} = \frac{T_i}{2} + MTTR \tag{4-8}$$

$$t_{C2} = MTTR \tag{4-9}$$

$$t_{CE}' = \frac{\lambda^{DU}\left(\frac{T_i}{2} + MTTR\right) + (\lambda^{DU} + \lambda^{SD})MTTR}{\lambda^{DU} + \lambda^{DU} + \lambda^{SD}} \tag{4-10}$$

Ainsi, la probabilité moyenne de défaillance sur demande est donnée par la formule suivante [CEI 06] :

$$PFDavg = (\lambda^{DU} + \quad \lambda^{DD} + \lambda^{SD})t_{CE}' \tag{4-11}$$

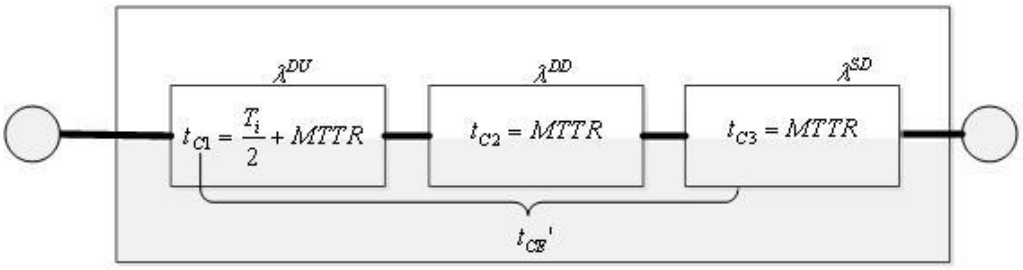



Figure 4-8 : Schéma du principe de la fiabilité pour une structure 1oo1D

Le calcul qui suit concerne le cas de la puce FPGA Spartan-3E, car elle répond aux contraintes du niveau de sécurité exigée.

Étant donné le taux de défaillance de base pour chaque famille FPGA, le taux de défaillance dangereuse détectée et le taux de défaillance dangereuse non détectée peuvent être déduites par les formules suivantes [CEI 06] :

$$\lambda^{DU} = \frac{\lambda}{2}(1-DC) \quad , \lambda^{DD} = \lambda^{SD} = \frac{\lambda}{2}DC \quad ; \quad \lambda^{D} = \frac{\lambda}{2} \tag{4-12}$$

La probabilité moyenne de défaillance sur demande de la fonction de sécurité pour les deux architectures du MIF 1oo1 et 1oo1D et pour un temps de mission Ti = 1 [an] est représentée sur le Tableau 4-9 :

Le MIF d'architecture 1oo1D avec diagnostic est 200 fois plus résistant aux défaillances dangereuses qu'un MIF de structure 1oo1 sans test de diagnostic qui agit directement sur la sortie.

On constate aussi que l'influence des tests de diagnostic (DC) sur la valeur de la probabilité moyenne de défaillance sur demande est minimale par rapport à l'impact de ce dernier sur la valeur PFDavg d'un MIF de structure 1oo1.

L'influence du temps de mission sur la probabilité de défaillance dangereuse est représentée sur le Tableau 4-10 :

| | La valeur PFDavg par [an] | |
|---|---|---|
| | Taux de défaillance Spartan-3E λ=397 E-09 par [h] | |
| **DC** | **MIF de structure 1oo1** | **MIF de structure 1oo1D** |
| **0 %** | **8,71** E-**04** | **3,18** E-**06** |
| **60 %** | **3,49** E-**04** | **4,13** E-**06** |
| **90 %** | **8,85** E-**05** | **4,61** E-**06** |
| **99 %** | **1,03** E-**05** | **4,75** E-**06** |

Tableau 4-9 : L'impact du DC sur les valeurs PFDavg du MIF (1oo1) et (1oo1D)

| | | Probabilité moyenne de défaillance sur demande PFDavg | | | |
|---|---|---|---|---|---|
| **FPGA** | **DC** | **Ti = 1 an** | **Ti = 3 ans** | **Ti = 5 ans** | **Ti = 10 ans** |
| **Spartan-3E** | **0 %** | **3,18 E-06** | **3,18** E-**06** | **3,18** E-**06** | **3,18** E-**06** |
| | **60 %** | **4,10 E-06** | **4,13** E-**06** | **4,13** E-**06** | **4,13** E-**06** |
| | **90 %** | **4,61** E-**06** | **4,61** E-**06** | **4,61** E-**06** | **4,61** E-**06** |
| | **99 %** | **4,75** E-**06** | **4,75** E-**06** | **4,75** E-**06** | **4,75** E-**06** |

Tableau 4-10 : Impact de Ti sur la valeur PFD sur le MIF d'une structure 1oo1D

## 4.5 Moteur d'inférence floue d'une architecture redondante

Le moteur d'inférence floue avec une architecture redondante au moins un parmi deux (1oo2) possède deux moteurs d'inférence floue MIF1 et MIF2 d'une structure similaire (Figure 4-9). Dans ce système, la défaillance d'un moteur d'inférence floue n'empêche donc pas l'exécution de la fonction de sécurité, car elle pourra être exécutée par le canal non défaillant. Les tests de diagnostics sont présents seulement pour la réparation du système. Donc le système sera défaillant lorsque les deux MIF seront défaillants.

Le point fort de cette architecture est qu'elle ne possède qu'une faible probabilité de défaillance à la demande, car la probabilité qu'une défaillance dangereuse se produise en même temps et qu'elle fasse partie de la même famille d'erreurs quand les deux systèmes d'inférence floue tendent vers zéro. Chaque MIF du système possède des tests de diagnostics et les résultats des deux MIF sont contrôlés par le module de la comparaison.

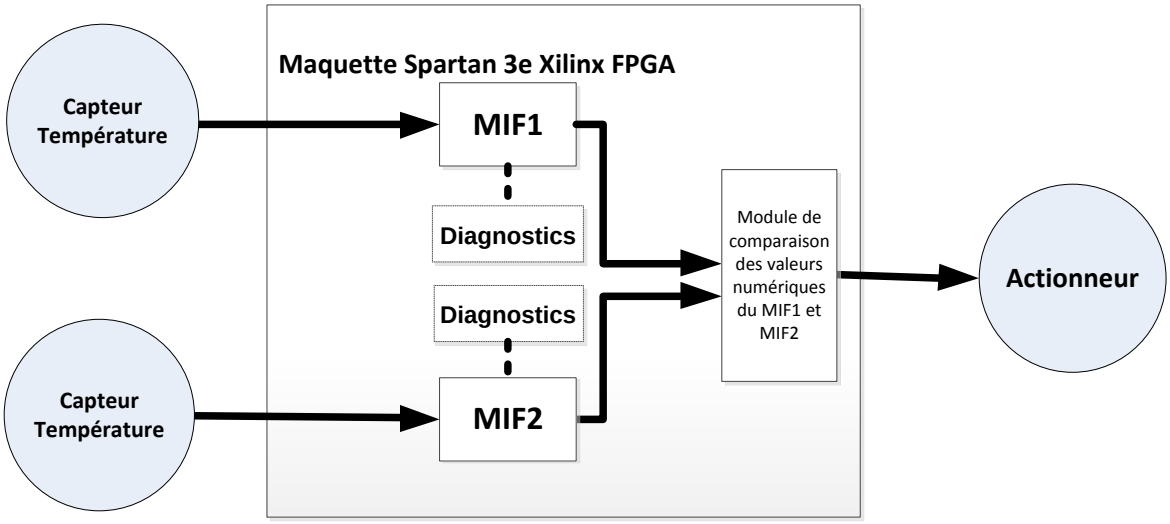



Figure 4-9 : MIF d'une architecture 1oo2

Le Tableau 4-11 montre les différents états du système 1oo2. Par définition, la fonction de sécurité se résume à la rupture de la tension de sortie après la fin du temps de réparation MTTR en cas d'une détection d'une défaillance dangereuse au niveau de canal et au niveau des deux canaux.

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **État du système** | **Cause potentielle** | **Disponibilité du système** | **Sécurité de système** |
| Un canal est défaillant. | Une défaillance dangereuse (voir le Tableau 4-1 et le Tableau 4-2). | Possibilité d'actionner la sortie par le canal non défaillant. La disponibilité d'assurer la fonction de sécurité est satisfaite. | Le système est sûr, puisque la fonction de sécurité peut être assurée par le canal non défaillant. |
| Deux canaux sont défaillants. | | Impossibilité d'actionner la sortie. La disponibilité d'assurer la fonction de sécurité n'est pas satisfaite. | Le système n'est pas fiable. |

Tableau 4-11 : Disponibilité et la sécurité dans une architecture 1oo2

Le temps de l'indisponibilité $t_{CE}$ d'un canal à cause d'une défaillance dangereuse détecté est représenté par [CEI 06] :

$$t_{CE} = \frac{\lambda^{DU}}{\lambda^{D}}(\frac{T_i}{2} + MTTR) + \frac{\lambda^{DD}}{\lambda^{D}} MTTR \quad (4\text{-}13).$$

On ajoute aussi le temps de l'indisponibilité de l'autre canal à cause d'une défaillance dangereuse détectée qui est représentée par $t_{GE}$ [CEI 06] :

$$t_{GE} = \frac{\lambda^{DU}}{\lambda^{D}}(\frac{T_i}{3} + MTTR) + \frac{\lambda^{DD}}{\lambda^{D}} MTTR \quad (4\text{-}14).$$

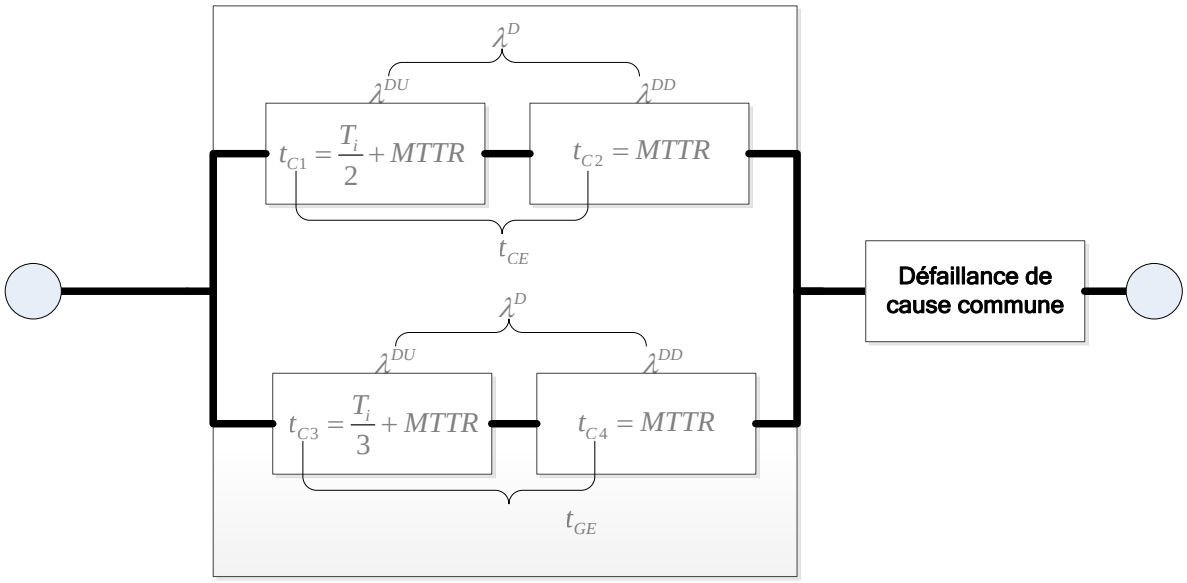



Figure 4-10 : Schéma du principe de la fiabilité pour une structure 1oo2

L'introduction des défaillances de mode commun est généralement modélisée par le facteur bêta, en tenant compte des défaillances couplées CMF (*common mode failure*). On obtient la formule [CEI 06] pour le calcul de la valeur PFD(Ti) pour une redondance homogène :

$$PFDavg = 2((1-\beta_D)\lambda^{DD} + (1-\beta)\lambda^{DU})^2 t_{CE} t_{GE} + \beta_D \lambda^{DD} MTTR + \beta\lambda^{DU}(\frac{T_i}{2} + MTTR) \quad (4\text{-}15).$$

avec le facteur bêta des défaillances couplées $\beta$

et le facteur bêta des défaillances couplées dangereuses $\beta_D$ .

Généralement, selon la norme de sécurité 61508 [CEI 06] les facteurs bêta $\beta$ et $\beta_D$ peuvent prendre des valeurs comprises entre 1 et 20 %.

L'influence du test de diagnostic et le facteur bêta de la sensibilité aux défaillances de cause commune sur la probabilité moyenne de défaillance sur demande de la fonction de sécurité pour un temps de mission Ti = 1 [an] est représentée sur le Tableau 4-12.

| **Architecture** | **La valeur PFDavg par [an]** | | | |
|---|---|---|---|---|
| **MIF 1oo1** | **DC** | **λ = 397 E-09 par [h]** | | |
| | | **$\beta$ = 2 %** **$\beta_D$ = 1 %** | **$\beta$ = 10 %** **$\beta_D$ = 5 %** | **$\beta$ = 20 %** **$\beta_D$ = 10 %** |
| | **0 %** | 8,71 E-04 | | |
| | **60 %** | 3,49 E-04 | | |
| | **90 %** | 8,85 E-05 | | |
| | **99 %** | 1,03 E-05 | | |
| **MIF 1oo1D** | **0 %** | 3,18 E-06 | | |
| | **60 %** | 4,13 E-06 | | |
| | **90 %** | 4,61 E-06 | | |
| | **99 %** | 4,75 E-06 | | |
| **MIF 1oo2** | **0 %** | 4,45 E-06 | 2,20 E-05 | 4,39 E-05 |
| | **60 %** | 1,76 E-06 | 8,78 E-06 | 1,76 E-05 |
| | **90 %** | 4,4 E-07 | 2,21 E-06 | 4,42 E-06 |
| | **99 %** | 4,78 E-08 | 2,39 E-07 | 4,78 E-07 |

Tableau 4-12 : L'impact du facteur DC sur les valeurs PFDavg du MIF (1oo2)

Le MIF d'architecture redondante 1oo2 avec diagnostic est 5 fois plus résistant aux défaillances dangereuses qu'un MIF de structure 1oo1.

On constate aussi que l'influence des tests de diagnostic (DC) sur la valeur de la probabilité moyenne de défaillance sur demande est grande. Un MIF de structure redondante 1oo2 avec une couverture de diagnostic DC = 99 % est 2 fois plus résistant aux défaillances dangereuses qu'un MIF de la même structure avec un facteur DC égal à 60 %.

On peut constater aussi une influence du facteur bêta de la sensibilité aux défaillances de cause commune sur la probabilité moyenne de défaillance sur demande. Un MIF

d'architecture redondante dont les facteurs bêta $\beta$ = 20 % et $\beta_D$ = 10 % est 10 fois plus sensible a des défaillances dangereuses qu'un MIF dont les facteurs bêta sont égaux à $\beta$ = 2 % et $\beta_D$ = 1 %.

L'influence du temps de mission sur la probabilité de défaillance dangereuse est présentée par le Tableau 4-13 :

| **Architecture** | **La valeur PFDavg** | | | | |
|---|---|---|---|---|---|
| **MIF 1oo2** | **DC** | **λ=397 E-09 par [h]** $\beta$ **=2 %** $\beta_D$ **=1 %** | | | |
| | | **Ti = 1 an** | **Ti = 3 ans** | **Ti = 5 ans** | **Ti = 10 ans** |
| | **0 %** | 1,84 E-05 | 6,09 E-05 | 1,11 E-04 | 2,71 E-04 |
| | **60 %** | 7,12 E-06 | 2,23 E-05 | 3,87 E-05 | 8,53 E-05 |
| | **90 %** | 1,76 E-06 | 5,32 E-06 | 8,96 E-06 | 1,84 E-05 |
| | **99 %** | 1,90 E-07 | 5,38 E-07 | 8,88 E-07 | 1,76 E-06 |

Tableau 4-13 : L'impact du Ti sur les valeurs PFDavg du MIF de structure 1oo2

Si on retarde le temps de mission Ti de 10 ans pour ce genre d'architecture 1oo2 du MIF implémenté dans la puce FPGA de la famille Spartan-3E, alors le risque que notre moteur d'inférence floue subisse une défaillance dangereuse est multiplié par un facteur de 10 (voir Tableau 4-13).

## 4.6 Moteur d'inférence floue d'une architecture redondante avec diagnostic

Le moteur d'inférence floue avec une architecture redondante 1oo2D

(

Figure 4-11) est similaire à l'architecture redondante 1oo2, mais avec un plus, c'est que les tests de diagnostic peuvent agir sur la sortie de MIF dans le cas d'une détection d'une défaillance dangereuse. Le système est défaillant quand les deux contrôleurs flous sont défaillants.

Le Tableau 4-14 montre les différents états du système 1oo2D. Par définition, la fonction de sécurité se résume à la rupture de la tension de sortie après la fin du temps de réparation MTTR en cas d'une détection d'une défaillance dangereuse au niveau d'un canal et au niveau des deux canaux.

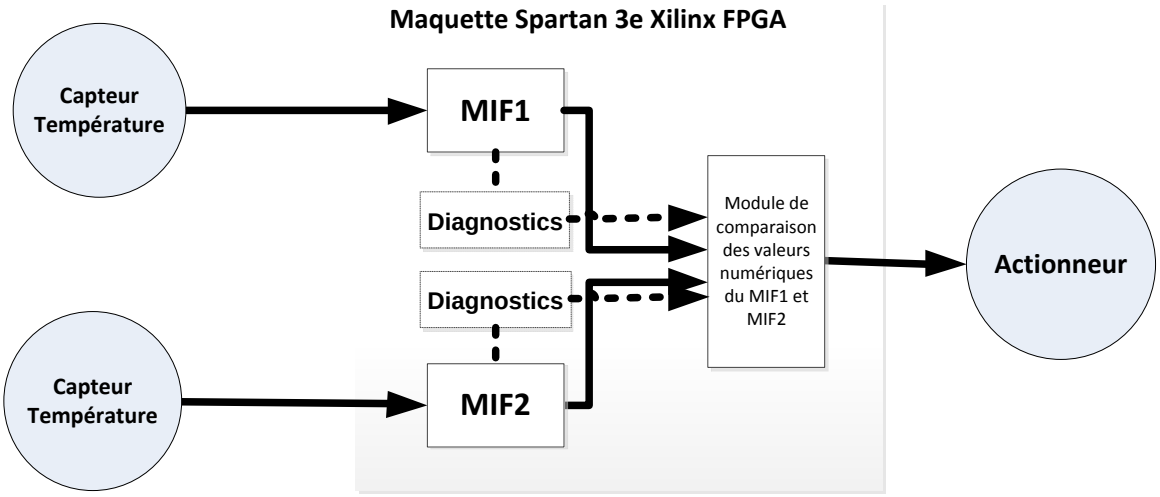



Figure 4-11 : MIF d'une architecture 1oo2D

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **État du système** | **Causes potentielles** | **Disponibilité du système** | **Sécurité du système** |
| Un canal est défaillant | Une défaillance dangereuse | <u>Défaillance détectée et non détectée.</u><br>Possibilité d'actionner la sortie par le canal non défaillant. La disponibilité de continuer d'assurer la fonction de sécurité est satisfaite. | |
| Deux canaux défaillants | | <u>Défaillance détectée.</u><br>Possibilité d'actionner la sortie par les tests de diagnostic, mais la disponibilité de continuer d'assurer la fonction de sécurité n'est pas satisfaite. | Le système est sûr. La fonction de sécurité est toujours assurée par les tests de diagnostic. |
| | | <u>Défaillance non détectée.</u><br>Impossibilité d'actionner la sortie. | Le système n'est plus sûr. |

Tableau 4-14 : Disponibilité et la sécurité dans une architecture 1oo2D

Le temps de l'indisponibilité d'un canal à cause d'une défaillance dangereuse détectée est représenté par $t_{CE}$' :

$$t_{CE}' = \frac{\lambda^{DU}\left(\frac{T_i}{2} + MTTR\right) + (\lambda^{DD} + \lambda^{SD})MTTR}{\lambda^{DD} + \lambda^{DU} + \lambda^{SD}} \quad (4\text{-}16)$$

On ajoute aussi le temps de l'indisponibilité de l'autre canal à cause d'une défaillance dangereuse non détectée qui est représentée par $t_{GE}'$ :

$$t_{GE}' = \frac{\lambda^{DU}\left(\frac{T_i}{3} + MTTR\right) + (\lambda^{DD} + \lambda^{SD})MTTR}{\lambda^{DD} + \lambda^{DU} + \lambda^{SD}} \quad (4\text{-}17)$$

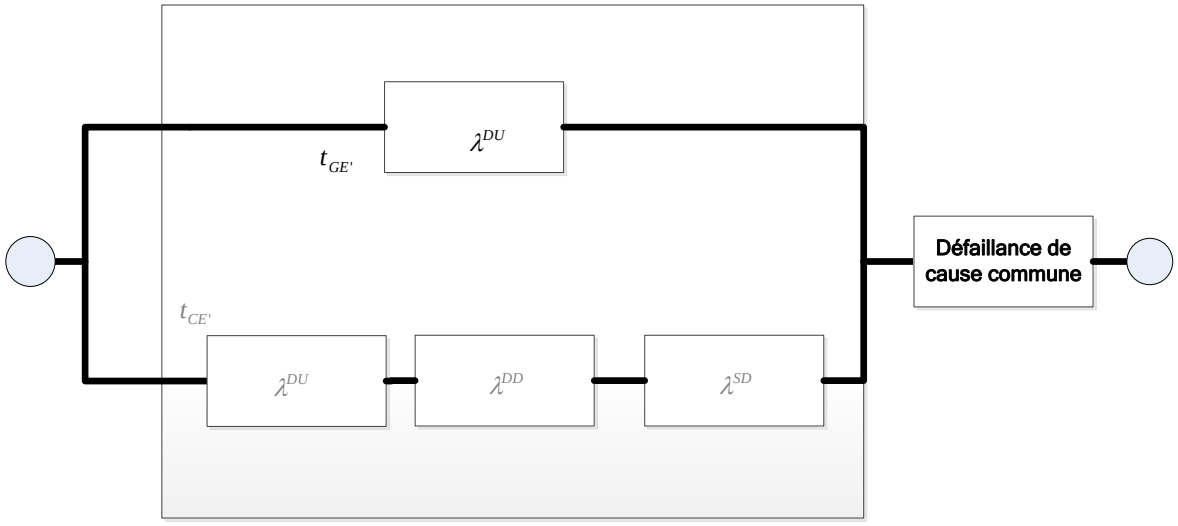



Figure 4-12 : Schéma du principe de la fiabilité pour une structure 1oo2D

De plus des défaillances de mode commun qui sont généralement modélisées par le facteur bêta, en tenant compte des défaillances couplées CMF (*common mode failure*), on ajoute les défaillances sûres détectées. On obtient la formule suivante pour le calcul de la valeur PFD(Ti) pour une redondance homogène :

$$PFDavg = 2(1-\beta)\,\lambda^{DU}((1-\beta)\lambda^{DU} + (1-\beta_D)\lambda^{DD} + \lambda^{SD})t_{CE'}t_{GE'} + \beta_D\lambda^{DD}MTTR + \beta\lambda^{DU}(\frac{T_i}{2} + MTTR) \quad (4\text{-}18)$$

L'influence du test de diagnostic et le facteur bêta de la sensibilité aux défaillances de cause commune sur la probabilité moyenne de défaillance sur demande de la fonction de sécurité pour un temps de mission Ti = 1 [an] est présentée sur le Tableau 4-15 :

| **Architecture** | **La valeur PFDavg par [an]** | | | |
|---|---|---|---|---|
| **MIF 1oo2** | **DC** | **λ=397 E-09 par [h]** | | |
| | | **$\beta$ =2 %** **$\beta_D$ =1 %** | **$\beta$ =10 %** **$\beta_D$ =5 %** | **$\beta$ =20 %** **$\beta_D$ =10 %** |
| | **0 %** | 4,45 E-06 | 2,20 E-05 | 4,39 E-05 |
| | **60 %** | 1,76 E-06 | 8,78 E-06 | 1,76 E-05 |
| | **90 %** | 4,42 E-07 | 2,21 E-06 | 4,42 E-06 |
| | **99 %** | 4,78 E-08 | 2,39 E-07 | 4,78 E-07 |
| **MIF 1oo2D** | **0 %** | 1,84 E-05 | 8,79 E-05 | 1,75 E-04 |
| | **60 %** | 6,85 E-08 | 1,78 E-07 | 3,16 E-07 |
| | **90 %** | 1,81 E-08 | 8,83 E-08 | 1,76 E-07 |
| | **99 %** | 1,60 E-08 | 8,02 E-08 | 1,60 E-07 |

Tableau 4-15 : L'impact du DC sur les valeurs PFDavg du MIF (1oo2 et 1oo2D)

L'impact du facteur bêta de la sensibilité aux défaillances de cause commune sur la probabilité moyenne de défaillance sur demande comme le montre le Tableau 4-15 est très important.

L'influence du temps de mission Ti sur la probabilité de défaillance dangereuse sur le MIF d'architecture 1oo2D est présentée dans le Tableau 4-16.

Si on retarde le temps de mission Ti de 10 ans pour ce genre d'architecture redondante 1oo2D du MIF implémenté dans la puce FPGA de la famille Spartan-3E, le risque que notre moteur d'inférence floue subisse une défaillance dangereuse se multiplie par un facteur de 57 pour une couverture de diagnostic de DC = 60 %.

| Architecture | La valeur PFDavg | | | | |
|---|---|---|---|---|---|
| MIF 1oo2D | DC | λ = 397 E-09 par [h]<br>$\beta$ = 2 %<br>$\beta_D$ = 1 % | | | |
| | | Ti=1an | Ti=3ans | Ti=5ans | Ti=10ans |
| | 0 % | 1,84 E-05 | 6,09 E-05 | 1,11 E-04 | 2,71 E-04 |
| | 60 % | 6,85 E-08 | 3,83 E-07 | 1,01 E-06 | 3,95 E-06 |
| | 90 % | 1,81 E-08 | 2,24 E-08 | 3,07 E-08 | 6,97 E-08 |
| | 99 % | 1,60 E-08 | 1,60 E-08 | 1,61 E-08 | 1,61 E-08 |

Tableau 4-16 : L'impact du Ti sur les valeurs PFDavg du MIF de structure 1oo2

## 4.7 Moteur d'inférence floue d'une architecture redondante 2oo2

Un moteur d'inférence floue d'architecture 2oo2 consiste en deux systèmes d'inférences floues, ce qui donne une architecture similaire à l'architecture 1oo2, mais cette structure du MIF, ne tolère pas une défaillance dangereuse au niveau d'un canal. Dans le cas d'une défaillance dangereuse dans un MIF, le système est dans un état dangereux et ne peut pas exécuter la fonction de sécurité. La Figure 4-13 présente la structure du MIF 2oo2. Les tests de diagnostics sont présents seulement pour la réparation du système.

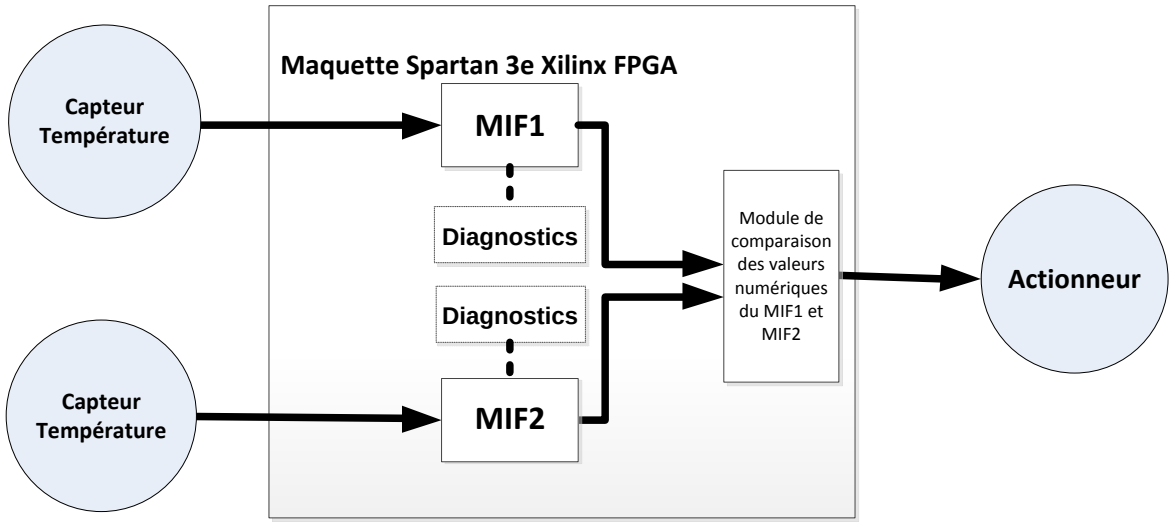



Figure 4-13 : MIF d'une architecture 2oo2

On remarque une similitude entre l'architecture 2oo2 et la structure 1oo2. En revanche, au niveau du fonctionnement, il est impossible de continuer à exécuter la fonction de sécurité par un autre canal lorsqu'une défaillance dangereuse est détectée dans un canal. Autrement dit, une défaillance dangereuse d'un seul canal entraînera la défaillance du système.

Le Tableau 4-17 montre les différents états du système d'architecture 2oo2. Par définition, la fonction de sécurité se résume à la rupture de la tension de sortie après la fin du temps de réparation MTTR en cas d'une détection d'une défaillance dangereuse au niveau d'un canal ou bien niveau des deux canaux.

| **La fonction de sécurité est assurée si la sortie n'est plus alimentée** | | | |
|---|---|---|---|
| **État du système** | **Causes potentielles** | **Disponibilité du système** | **Sécurité de système** |
| Un canal est défaillant ou bien les deux canaux sont défaillants. | Une défaillance dangereuse (voir le Tableau 4-1 et le Tableau 4-2) | Impossibilité d'actionner la sortie par le canal non défaillant. La disponibilité de continuer d'assurer la fonction de sécurité n'est pas satisfaite. | Le système n'est pas sûr, puisque la fonction de sécurité ne peut pas être assurée. |

Tableau 4-17 : La disponibilité et la sécurité dans une architecture 2oo2

Le temps d'indisponibilité équivaut à deux fois le temps d'indisponibilité pour un canal $t_{CE}$ (Figure 4-14) :

$$t_{CE} = \frac{\lambda^{DU}}{\lambda^{D}}\left(\frac{T_i}{2} + MTTR\right) + \frac{\lambda^{DD}}{\lambda^{D}} MTTR \qquad (4\text{-}19)$$

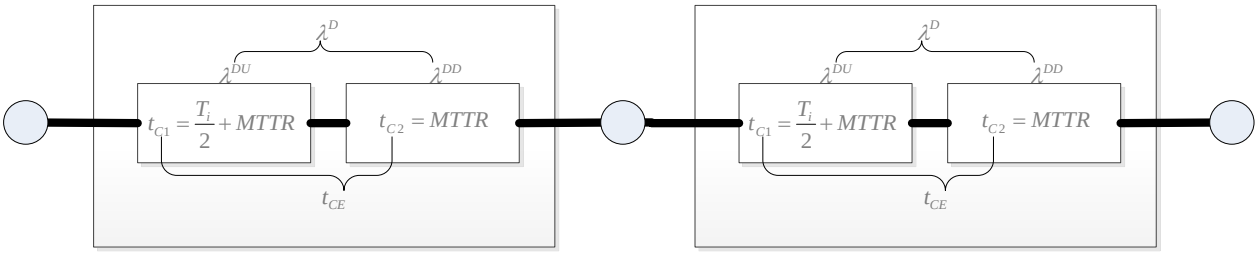



Figure 4-14 : Schéma du principe de la fiabilité pour une structure 2oo2

Ainsi, la probabilité moyenne de défaillance sur demande est :

$$PFDavg = 2(\lambda^{DU} + \lambda^{DD})t_{CE} \quad (4\text{-}20).$$

L'influence du test de diagnostic sur la probabilité moyenne de défaillance sur demande de la fonction de sécurité pour un temps de mission Ti = 1 [an] est représentée par le Tableau 4-18 :
D'après les valeurs de la probabilité moyenne de défaillance sur demande, on constate que plus le facteur de diagnostic (DC) augmente plus le risque que le MIF d'architecture 2oo2 subisse une défaillance dangereuse diminue, pour atteindre un facteur de minimisation du risque de 84 fois pour un facteur DC = 99 % comparé à celui de DC = 0 %. On constate aussi que le MIF de structure 1oo2 est en moyenne 100 fois plus robuste contre les défaillances dangereuses qu'un MIF de structure 2oo2.

| **Architecture** | **La valeur PFDavg par [an]** | | | |
|---|---|---|---|---|
| **MIF 2oo2** | **DC** | **λ = 397 E-09 par [h]** | | |
| | | **$\beta$ = 2 %** **$\beta_D$ = 1 %** | **$\beta$ = 10 %** **$\beta_D$ = 5 %** | **$\beta$ = 20 %** **$\beta_D$ = 10 %** |
| | **0 %** | 1,74 E-03 | | |
| | **60 %** | 6,99 E-04 | | |
| | **90 %** | 1,77 E-04 | | |
| | **99 %** | 2,06 E-05 | | |
| **MIF 1oo2** | **0 %** | 4,45 E-06 | 2,20 E-05 | 4,39 E-05 |
| | **60 %** | 1,76 E-06 | 8,78 E-06 | 1,76 E-05 |
| | **90 %** | 4,42 E-07 | 2,21 E-06 | 4,42 E-06 |
| | **99 %** | 4,78 E-08 | 2,39 E-07 | 4,78 E-07 |

Tableau 4-18 : L'impact du facteur DC sur les valeurs PFDavg du MIF (2oo2)

L'influence du temps de mission sur la probabilité de défaillance dangereuse est représentée sur le Tableau 4-19:

| Architecture | La valeur PFDavg | | | | |
|---|---|---|---|---|---|
| | DC | λ = 397 E-09 par [h] | | | |
| | | Ti = 1 an | Ti = 3 ans | Ti = 5 ans | Ti = 10 ans |
| MIF 2oo2 | 0 % | 1,74 E-03 | 5,22E-03 | 8,70 E-03 | 1,74 E-02 |
| | 60 % | 6,99 E-04 | 2,09 E-03 | 3,48 E-03 | 6,96 E-03 |
| | 90 % | 1,77 E-04 | 5,25 E-04 | 8,73 E-04 | 1,74 E-03 |
| | 99 % | 2,06 E-05 | 5,53 E-05 | 9,01 E-05 | 1,77 E-04 |

Tableau 4-19 : L'impact du Ti sur les valeurs PFDavg du MIF (2oo2)

Si l'on retarde le temps de mission Ti de 10 ans pour ce genre d'architecture redondante 2oo2 du MIF implémenté dans la puce FPGA de la famille Spartan-3E, le risque que notre moteur d'inférence floue subisse une défaillance dangereuse sera multiplié par un facteur de 10 pour une couverture de diagnostic de DC = 60 %.

## 4.8 Le choix de l'architecture du MIF

À partir des différentes valeurs des taux de défaillance, des différents temps de mission Ti, on analysera les valeurs probabilistes de défaillances sur demande pour les différentes architectures du MIF étudiées précédemment.

Les considérations suivantes seront prises en compte :

- Si la durée de test d'inspection (*proof test*) est de 10 ans, cela signifie que pour cette période, le test d'inspection hors ligne ne sera pas effectué. Les tests approfondis sont exécutés dès la mise sous tension du système. Les tests de diagnostics veillent ensuite sur le système.
- La durée de réparation MTTR (Tr) est égale à 8 heures.
- Le taux de couverture des tests de diagnostics est de 60 %.
- Les facteurs bêta pris en considération sont de 2 % et 1 %.
- Le temps de mission d'un an correspond à un service de 8 760 heures.
- Le taux de défaillance considéré pour une structure homogène du MIF mise en œuvre dans une puce FPGA est de $1{,}19^{E}$ -07 [1/heure] pour les défaillances non détectées et d'un taux de défaillance de $7{,}94^{E}$ -08 pour les défaillances détectées. Ces valeurs ont été prises à partir des données fournies par la norme Siemens 95000 [SSN 86].

Les probabilités de défaillances dangereuses seront données par an. La probabilité que le moteur d'inférence floue ne puisse pas exécuter la fonction de sécurité pour laquelle il

a été conçu au moment où la demande de cette fonction est représentée par le Tableau 4-20.

À partir du Tableau 4-20, on peut constater d'une part que les valeurs de la probabilité de défaillances sur demande PFDavg des architectures avec diagnostics et sans diagnostics ne sont pas similaires et surtout pour une couverture de diagnostic de DC = 60 %. On peut constater qu'un MIF avec diagnostic est 50 fois plus résistant à une défaillance dangereuse qu'un MIF sans test de diagnostic.

| Architecture MIF | La valeur PFDavg par [an] | | | |
|---|---|---|---|---|
| | DC | λ = 397 E-09 par [h] | | |
| | | $\beta$ = 2 %<br>$\beta_D$ = 1 % | $\beta$ = 10 %<br>$\beta_D$ = 5 % | $\beta$ = 20 %<br>$\beta_D$ = 10 % |
| 1oo1 | 0 % | 8,71 E-04 | | |
| | 60 % | 3,49 E-04 | | |
| | 90 % | 8,85 E-05 | | |
| | 99 % | 1,03 E-05 | | |
| 1oo1D | 0 % | 3,18 E-06 | | |
| | 60 % | 4,13 E-06 | | |
| | 90 % | 4,61 E-06 | | |
| | 99 % | 4,75 E-06 | | |
| 1oo2 | 0 % | 4,45 E-06 | 2,20 E-05 | 4,39 E-05 |
| | 60 % | 1,76 E-06 | 8,78 E-06 | 1,76 E-05 |
| | 90 % | 4,42 E-07 | 2,21 E-06 | 4,42 E-06 |
| | 99 % | 4,78 E-08 | 2,39 E-07 | 4,78 E-07 |
| 2oo2 | 0 % | 1,74 E-03 | | |
| | 60 % | 6,99 E-04 | | |
| | 90 % | 1,77 E-04 | | |
| | 99 % | 2,06 E-05 | | |
| 1oo2D | 0 % | 1,84 E-05 | 8,79 E-05 | 1,75 E-04 |
| | 60 % | 6,85 E-08 | 1,78 E-07 | 3,16 E-07 |
| | 90 % | 1,81 E-08 | 8,83 E-08 | 1,76 E-07 |
| | 99 % | 1,60 E-08 | 8,02 E-08 | 1,60 E-07 |

Tableau 4-20 : La valeur PFDavg pour de différentes architectures

D'autre part, la meilleure valeur PFD est obtenue par l'architecture 1oo2D avec diagnostics d'une valeur égale à 1,6 E-08 et la mauvaise valeur a été obtenue par un MIF de structure de 2oo2 de 1,74 E-03.

À la lumière des différentes comparaisons des valeurs probabilistes obtenues pour différentes architectures du MIF (Tableau 4-20), les conclusions suivantes pour le moteur d'inférence floue ont été retenues :

- Le moteur d'inférence floue de structure 1oo1 est 49 fois plus confronté à une défaillance dangereuse qu'une architecture redondante 1oo2.
- Le moteur d'inférence floue de structure 1oo2D est 100 fois moins confronté à une défaillance dangereuse qu'une architecture redondante 1oo2.

## 4.9 Conclusion

L'analyse des différentes architectures a montré que le niveau de sécurité du moteur d'inférence floue dépend non seulement du taux de défaillance du composant, mais aussi du facteur de couverture du diagnostic qui indique à quel point le MIF est capable d'autosurveillance face à une éventuelle défaillance dangereuse, et de la nécessité d'une évaluation d'une éventuelle défaillance de cause commune qui peut invalider la redondance du MIF.

On a aussi constaté que la fréquence du test d'inspection a un impact négatif si on le retarde d'une période de plus de 10 ans, et que la meilleure valeur de la probabilité moyenne de défaillance sur demande de la fonction de sécurité a été obtenue pour l'architecture 1oo2D.

Il faut noter que l'architecture 1oo2D du MIF possède une tolérance minimale (HFT) à une faute dangereuse de 1. Ainsi, le MIF d'une telle architecture continue à assumer la fonction de sécurité requise en présence d'une anomalie dangereuse dans le système.

Dans le chapitre suivant seront présentés les tests de diagnostics utilisés dans le MIF d'une structure redondante 1oo2D, ainsi que les tests d'inspection mis en œuvre pour le régulateur MIF avec sécurité.

Le calcul des différentes valeurs PFDavg et MTTF est effectué par notre propre calcul. Le calcul utilise les taux de pannes de base du composant, le taux de couverture DC, le modèle du facteur β, le temps de mission Ti, une durée moyenne de rétablissement (MTTR) et les formules probabilistes [CEI 06] pour déterminer la valeur PFDavg pour les architectures 1oo1, 1oo1D, 1oo2, 1oo2, 1oo2D et 2oo2.

# Chapitre 5

# Modélisation quantitative du moteur d’inférence floue sûr d’architecture (1oo2D)

## *Résumé du chapitre 5 :*

*Ce chapitre est consacré à la modélisation quantitative du MIF. Nous faisons une analyse détaillée des modes de défaillances et effets au niveau des composants qui constituent le régulateur flou d'architecture (1oo2D) : Convertisseur analogique-numérique AD, convertisseur numérique-analogique DA, moteur d'inférence floue implémenté en FPGA et le circuit d'alimentation, ceci pour pouvoir faire la quantification du moteur d'inférence floue par le calcul de la probabilité moyenne de défaillance sur demande PFDavg par la méthode de l'arbre des causes, par la méthode du bloc-diagramme de fiabilité et par la méthode des chaînes de Markov.*

# 5 Modélisation quantitative du moteur d'inférence floue sûr (MIFS)

## 5.1 Introduction

L'évaluation des performances du MIF peut s'obtenir par des méthodes quantitatives. Cette évaluation s'apparente à un calcul d'indisponibilité de la fonction de sécurité lors de sa sollicitation, à partir des données probabilistes de défaillances, [YFR 01], [GBL 01].

Dans ce cadre, les blocs-diagrammes de fiabilité apportent une bonne formalisation du système en présentant une approche plus fonctionnelle du MIF. L'évaluation de la probabilité moyenne de défaillance par la méthode de l'arbre des causes permet un modèle du MIF statique basé sur la détermination des causes amenant à l'événement dangereux. Par contre, l'évaluation des performances du MIF par les chaînes de Markov donne une allure très profonde sur les états que peut prendre le MIF en fonction des événements rencontrés (défaillances, test) et des paramètres étudiés (taux de défaillances, défaillances de causes communes).

La Figure 5-1 représente le moteur d'inférence floue avec sécurité d'architecture redondante (1oo2D) et qui est formé par les composants suivants :

- Traitement d'horloge.
- Module d'interface série maître SPI pour acquérir les valeurs digitalisées des canaux AI0 et AI1 de la part du convertisseur analogique-numérique A/D 14-bit de la société Linear Technologie LTC6912-1 [XSK 11]. Les valeurs sont enregistrées dans les registres ADC_Value_AI0 et ADC_Value_AI1 (Figure 5-1).
- Deux moteurs d'inférence floue.
- Deux modules « chien de garde ».
- Module d'interface série maître SPI pour contrôler la sortie analogique AO0 du convertisseur numérique-analogique D/A 12-bits de la société Linear Technologie [XSK 11].

La valeur de la probabilité de défaillance sur demande du moteur d'inférence floue sûr pour ces différents composants devra tout d'abord être calculée par la méthode des blocs-diagrammes de fiabilité, par la méthode de l'arbre des causes et ensuite par les chaînes de Markov.

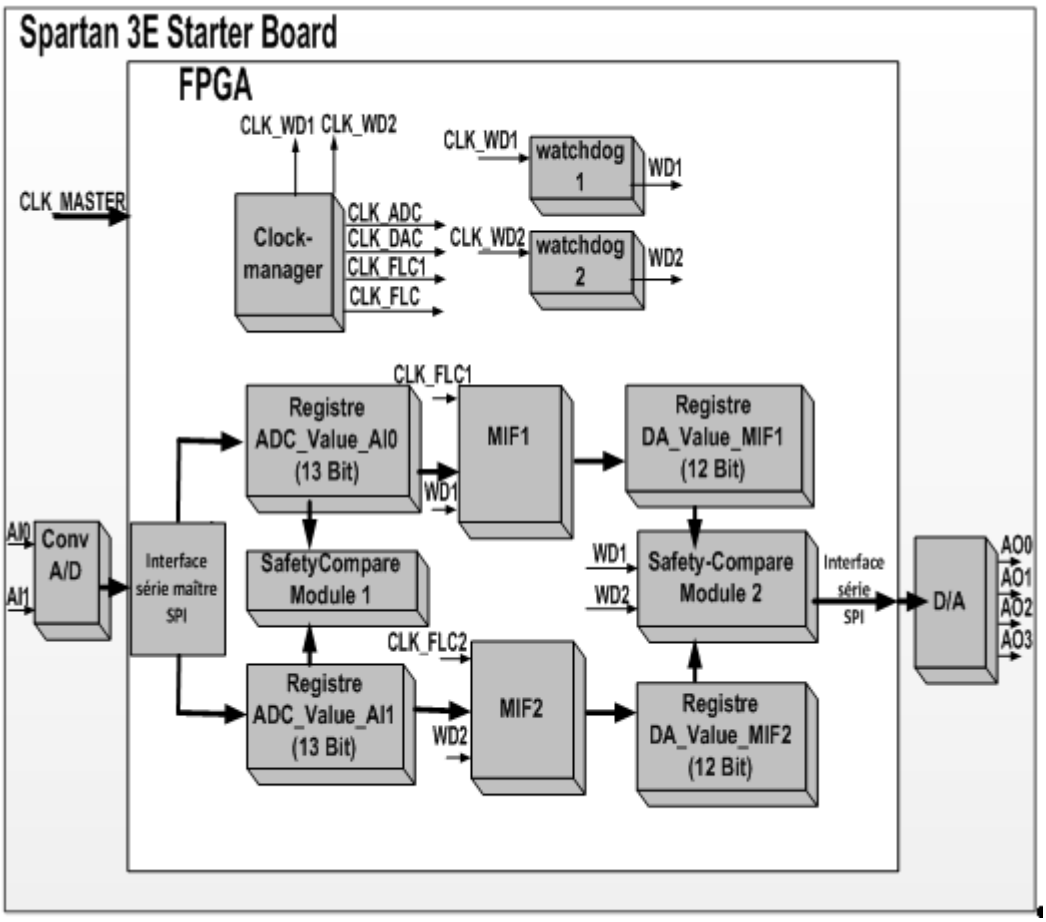



Figure 5-1 : Le moteur d'inférence floue à sécurité d'architecture 1oo2D

La complexité de la technologie FPGA ainsi que du système à implémenter nécessite une maîtrise d'implantation des tests de diagnostics afin qu'ils soient capables de détecter des défaillances avec un taux de couverture prédéfini. Pour ce faire, une analyse des modes de défaillance et effets (AMDEC) sera effectuée et présentée dans ce chapitre.

## 5.2 Analyse des modes de défaillances et effets du MIFS

L'analyse des modes de défaillances et effets est réalisée au niveau des composants suivants :

- convertisseur analogique-numérique ADC ;
- convertisseur numérique-analogique DAC ;
- moteur d'inférence floue implémenté en FPGA ;
- circuit d'alimentation.

Les erreurs potentielles dues à des fautes de court-circuit, de blocage et de retard sont observées dans l'analyse AMDEC.

### 5.2.1 Analyse AMDEC pour le convertisseur (ADC)

La chaîne « convertisseur analogique » se compose d'un amplificateur de la famille LTC69124 et d'un convertisseur analogique de la famille LTC1407/LTC1407A d'une résolution de 14 bits (Figure 5-3). Les deux entrées différentielles séparées sont simultanément échantillonnées sur le front montant du signal CONV (Figure 5-2). Après avoir été échantillonnées, elles sont converties à un taux de 1,5 Msps par canal. Le convertisseur possède une structure redondante *1 out of 2* avec un facteur de sécurité d'une valeur de 50 %, ce qui le rend de type A.

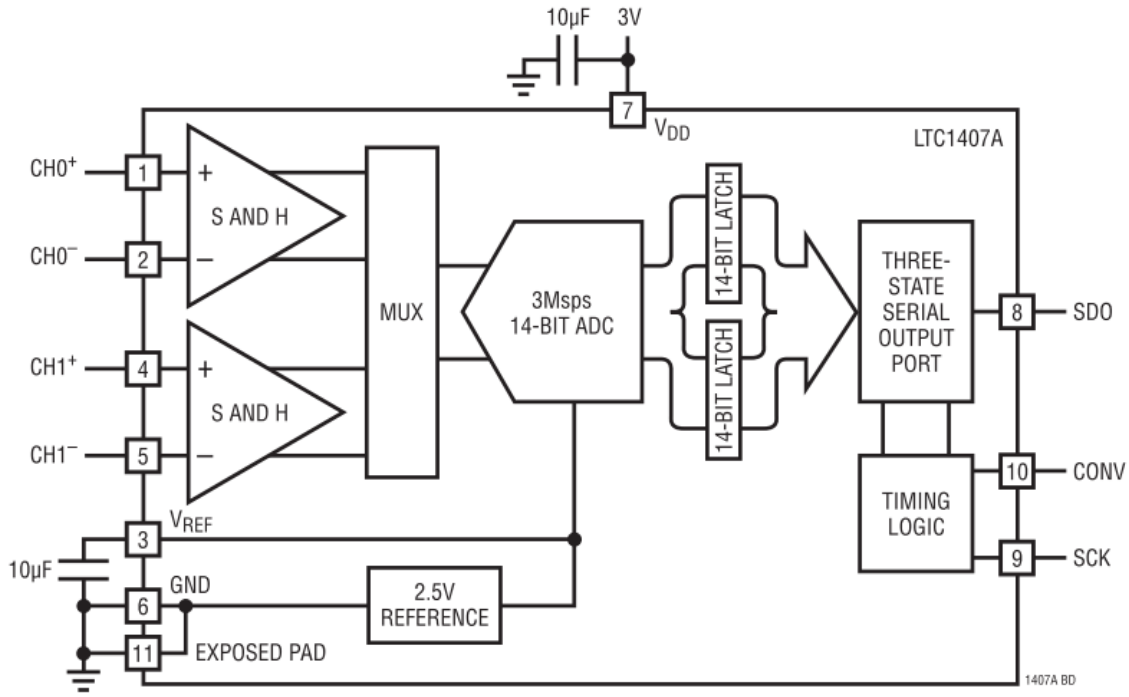



Figure 5-2 : La structure interne du convertisseur ADC [ADC 00]

Les valeurs d'entrées sont examinées d'une part au niveau des valeurs limites des signaux analogiques d'entrées, et d'autre part par la comparaison des valeurs digitales acquises de ces derniers. Il faut noter que la sortie du convertisseur ADC n'est pas isolée et n'a pas d'alimentation propre. Un blocage au niveau de sa sortie est possible.

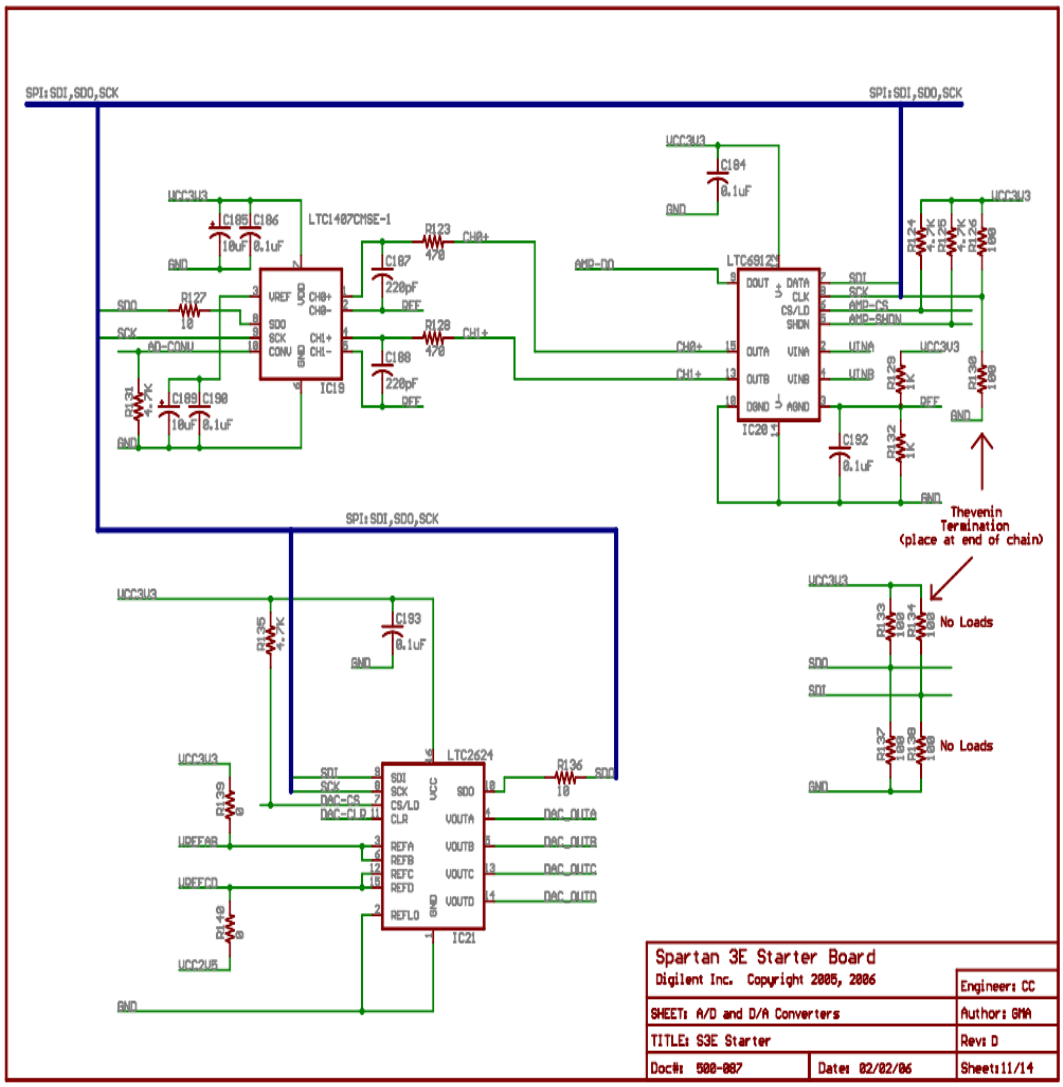



Figure 5-3 : Circuit électronique du convertisseur ADC et DAC [XSK 11]

Avant de commencer notre analyse des modes de défaillance et effets (AMDEC) du convertisseur ADC, il faut d'abord définir les entrées et les sorties de la puce de ce dernier :

| Alimentation | Entrées | Sorties |
|---|---|---|
| VCC3V3+ | CH0+ ; CH1+ | SDO |
| | CH0- ; CH1- | |
| | VREF | |
| | SCK | |
| | CONV | |

Tableau 5-1 : Les entrées et sorties du convertisseur ADC

L'AMDEC du convertisseur ADC est représentée par le Tableau 5-2 :

| **Erreurs potentielles** | **Causes potentielles** | **Défaillances potentielles** | **Diagnostic** |
|---|---|---|---|
| Court-circuit au niveau VCC3V3+ | Blocage *stuck-at-low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit au niveau CH0+/CH1+ | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit au niveau CH0-/CH1- | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit au niveau VREF | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit entre les entrées CH0+/CH0- | Blocage au niveau d'entrées | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit entre les entrées CH1+/CH1- | Blocage au niveau d'entrées | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit entre les entrées CH0-/CH1+ | Blocage au niveau d'entrées | Faute de mesure | Test de diagnostic TD1<br>Examen périodique des valeurs limites du convertisseur |
| Court-circuit au niveau de la sortie SDO | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Ne peut pas être détecté |
| Court-circuit au niveau de l'horloge SCK | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD2<br>Test de chien de garde |

Tableau 5-2 : Modèle d'erreur du convertisseur ADC

### 5.2.2 Analyse AMDEC pour le convertisseur numérique-analogique DAC

La Figure 5-4 montre les entrées et les sorties du convertisseur numérique-analogique DAC LTC 2624 avec une résolution de 12 bits en quadrature. Cela signifie qu'il dispose de quatre sorties analogiques indépendantes (A, B, C et D).

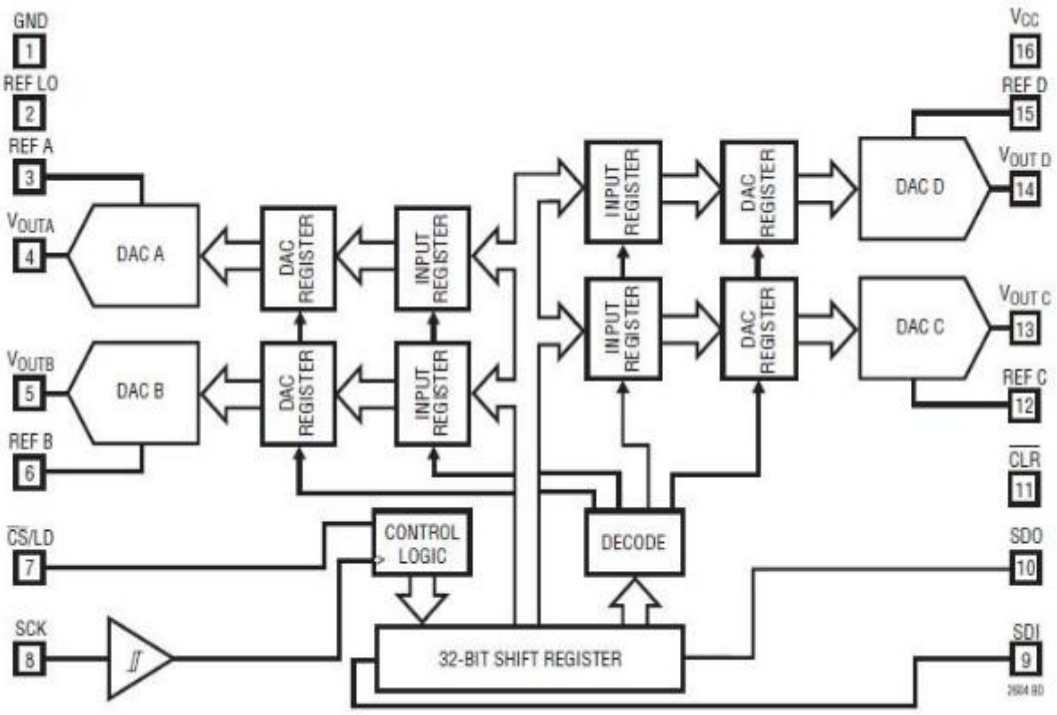



Figure 5-4 : Bloc-diagramme du convertisseur DAC [XSK 11]

Le Tableau 5-3 propose un résumé de ces entrées et sorties :

| Alimentation | Entrées | Sorties |
|---|---|---|
| VCC3V3+ | SDI | SDO |
| VREFA | CS | VOUTA |
| VREFB | SCK | VOUTB |
| VREFC | CLR | VOUTC |
| VREFD | SDO | VOUTD |

Tableau 5-3 : Les entrées et sorties du convertisseur DAC

L'AMDEC du convertisseur DAC est présentée par le Tableau 5-4 :

| **Erreur potentielle** | **Cause potentielle** | **Défaillance potentielle** | **Diagnostic** |
|---|---|---|---|
| Court-circuit au niveau VCC3V3+ | Blocage *stuck-at-low* (AGND) | Faute de mesure | Test de diagnostic TD1 |
| Court-circuit au niveau VREFA | Blocage *stuck-at-low* (AGND) | Faute de mesure | Test de diagnostic TD3 Comparaison entre les valeurs digitales de sortie VOUTA et VOUTB obtenues par les registres DACB et DACA |
| Court-circuit au niveau VREFB | Blocage *stuck-at-low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD3 Comparaison entre les valeurs digitales de sortie VOUTA et VOUTB obtenues par les registres DACB et DACA |
| Court-circuit au niveau SDI | Blocage *stuck-at-low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD4 Examen des données digitales pour la commande des sorties par la relecture de ces données par le registre FPGA_SDO au FPGA |
| Court-circuit au niveau CS | Blocage *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD5 Examen de la valeur précédente contre une chute brusque de la valeur mesure |
| Court-circuit au niveau de l'horloge SCK | Blocage *stuck-at low* (AGND) *stuck-at-high* (VCC3V3) | Faute de mesure | Test de diagnostic TD2 Test de chien de garde |

Tableau 5-4 : Modèle d'erreur du convertisseur DAC

### 5.2.3 Analyse AMDEC pour le régulateur flou MIFS

Les entrées et les sorties du moteur d'inférence floue sont représentées dans le Tableau 5-5 :

| Alimentation | Entrées | Sorties |
|---|---|---|
| VCC2V5 | In_SCK | Out_Data |
| | In_Data | Out_Done |
| | In_Start | |

Tableau 5-5 : Les entrées et sorties du MIF

L'AMDEC du MIFS est présentée par le
Tableau 5-6 :

| Erreur potentielle | Cause potentielle | Défaillance potentielle | Diagnostic |
|---|---|---|---|
| Court-circuit au niveau VCC2V5 | Blocage *stuck-at-low* (AGND) | Faute de mesure | Test de diagnostic TD1 |
| Court-circuit au niveau In_Data | | Faute de mesure | Test de diagnostic TD6. Comparaison entre les valeurs du premier et du deuxième MIF. Une discordance sera détectée. |
| Court-circuit au niveau InStart | | Faute de mesure | Test de diagnostic TD6. Comparaison entre les valeurs du premier et du deuxième MIF. Une discordance sera détectée. |
| Court-circuit au niveau In_SCK | | Faute de mesure | Test de diagnostic TD6. Comparaison entre les valeurs du premier et du deuxième MIF. Une discordance sera détectée. |

Tableau 5-6 : Modèle d'erreur du MIFS

## 5.3 Tests de diagnostics et les tests d'inspection

Les tests d'inspection sont exécutés dès que le système MIFS est mis sous tension, et contiennent les tests suivants :

- Le contrôle du contenu de la mémoire est effectué par la valeur contrôle de redondance cycle (CRC).

Le code VHDL du moteur d'inférence floue est protégé dans la mémoire SPI-Flash en ajoutant la valeur CRC à la fin du fichier MIFS.bit avant de le mémoriser. La valeur CRC est de longueur 16 bits. Elle est basée sur le fait que toute donnée binaire enregistrée dans la mémoire permet de construire un polynôme, chacun des bits donnant sa valeur au coefficient polynomial correspondant. Le polynôme générateur utilisé est un polynôme de degré 16 G(x) = x16 + x12 + x5 + 1.

- Le test du chien de garde.

Le chien de garde est relié par un générateur d'horloge interne en FPGA afin de surveiller l'horloge du système MIFS. Une disharmonie entre les deux horloges CLK_WD1 et CLK_WD2 sera détectée par la discordance des deux signaux gérés par les modules Watchdog1 et Watchdog2.

Les tests de diagnostics mis en œuvre dans le moteur d'inférence floue sûr sont définis dans le Tableau 5-7 :

| **Numéro du test de diagnostics** | **Défaillance potentielle** | **Description** |
|---|---|---|
| TD1 | Faute de mesure | Examen périodique des valeurs limites du convertisseur. |
| TD2 | Faute de mesure | Test du chien de garde. |
| TD3 | Faute de mesure | Comparaison entre les valeurs digitales de sortie VOUTA et VOUTB obtenues par les registres DACB et DACA. |
| TD4 | Faute de mesure | Examen des données digitales pour la commande des sorties par la relecture de ces données par le registre FPGA_SDO au FPGA. |
| TD5 | | Examen de la valeur précédente contre une chute brusque de la valeur mesurée. |
| TD6 | | Comparaison entre les valeurs du premier et du deuxième MIF. Une discordance sera détectée. |

Tableau 5-7 : Les tests de diagnostics du MIFS

## 5.4 Modélisation par bloc-diagramme de fiabilité

La modélisation du MIF par la méthode du bloc-diagramme de fiabilité permet le calcul de la disponibilité ou de la fiabilité du système. Tous les chemins entre l'entrée et la

sortie (voir dans les sous-chapitres suivants) décrivent les conditions pour que la fonction de sécurité soit accomplie. On suppose que les composants n'ont que deux états de fonctionnement (fonctionnement correct ou défectueux).

### 5.4.1 Fonction de sécurité du régulateur flou MIFS

La fonction de sécurité réalisée par le moteur d'inférence floue d'architecture redondante SFLC a pour but d'assurer ou de maintenir un état sûr du système par rapport à un événement dangereux spécifique comme des pannes aléatoires.

La fonction de sécurité consiste donc en une coupure de l'alimentation pour les 4 sorties analogiques du système. L'état sûr se met en place lorsque le MIFS détecte une défaillance due à un défaut du matériel comme les erreurs d'interconnexion, de blocages (*stuck-at-fault*), de transition, le déphasage de l'horloge ou encore une déviation de la valeur obtenue respectivement par les contrôleurs flous MIF1 et MIF2.

### 5.4.2 Architecture du moteur d'inférence flou MIFS

L'architecture du système doit être définie en fonction du design et de la décomposition fonctionnelle, comme cela est illustré sur la Figure 5-5 :

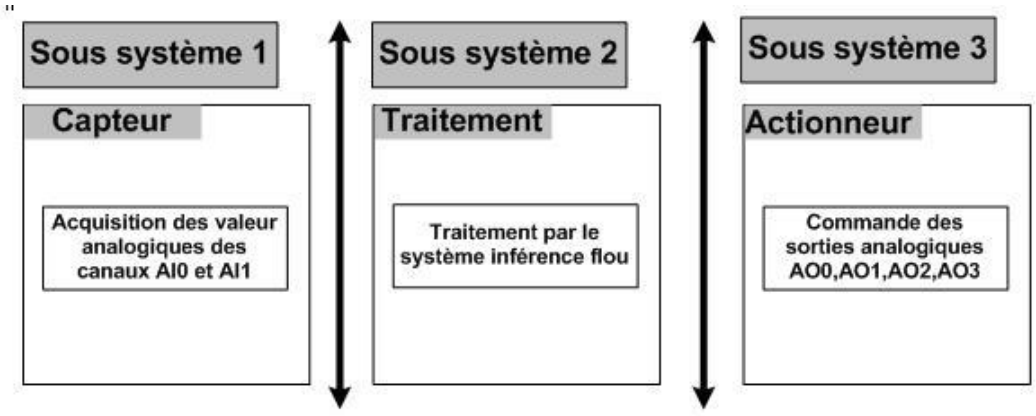



Figure 5-5 : Sous-systèmes de l'architecture du MIFS

la Figure 5-5 fournit le bloc-diagramme de fiabilité du moteur d'inférence floue sûr, constitué de trois sous-systèmes :

- La partie capteur au niveau de la carte en architecture 1oo1, composée d'alimentation, d'horloge de FPGA et le convertisseur analogique numérique avec son amplificateur et son filtre de passe-bas.
- La partie logique (traitement) en architecture 1oo2D, composée par deux contrôleurs flous homogènes et du chien de garde.

- La partie actionneur en architecture 1oo1, composée par le convertisseur numérique analogique.

### 5.4.3 Décomposition en blocs fonctionnels

Le moteur d'inférence floue contient 5 blocs fonctionnels. Les interconnexions entre les différents modules des sous-systèmes du MIFS sont représentées sur la figure 5-6 :

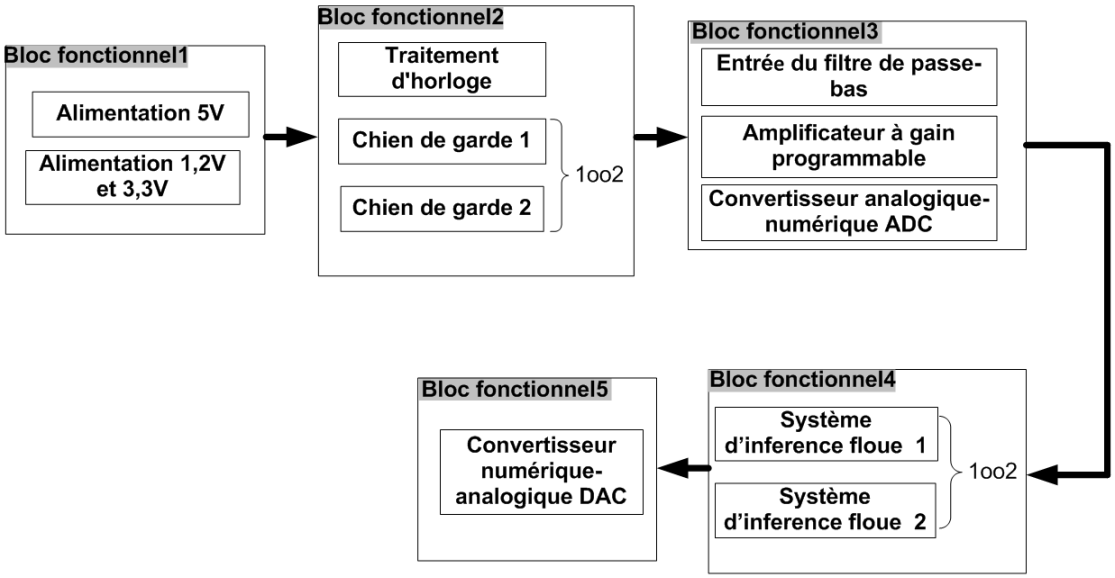



Figure 5-6 : Le bloc fonctionnel du moteur d'inférence floue

Les propriétés de chaque bloc fonctionnel (BF) sont représentées par le Tableau 5-8 :

| Sous-système | Bloc fonctionnel | Entrée | Sortie | Fonctionnalité |
|---|---|---|---|---|
| Acquisition des signaux du capteur | Alimentation (1oo1). | Tension d'une valeur de 5 V. | Différentes tensions (1,2 V, 3,3 V et 2,5 V). | Alimentation des éléments électroniques sur la carte Spartan-3E. |
| | Horloge (FPGA) (1oo1). | Fréquence de 50 MHz. | Différentes fréquences. | Division de la fréquence principale de 50 MHz à des fréquences définies. |
| | Convertisseur analogique ADC (1oo1). | Deux canaux analogiques d'une tension de 0,4 V-2,80 V. | Deux valeurs numérisées se situent dans l'intervalle 819 6…+8196. | Numérisation de la valeur analogique soumise aux canaux AI0 et AI1. S'il y a une discordance entre les deux valeurs numérisées, il est nécessaire d'arrêter la commande des sorties analogiques. |
| Traitement logique | Moteur d'inférence floue. (1oo2). | Deux valeurs numérisées qui se situent dans un intervalle 819 6…+8196. | Deux valeurs numérisées qui se situent dans un intervalle 0…4 095. | Moteur d'inférence floue avec structure redondante. S'il y a une discordance entre les deux valeurs numérisées, il est nécessaire d'arrêter la commande des sorties analogiques. |
| | Chien de garde (1oo2). | La fréquence CLK_WD1 et CLK_WD2 d'une valeur de 150 MHz. | Deux signaux internes WD1 et WD2, activés en état « high ». | Remise à l'état sûr, portant la coupure de l'alimentation au niveau de l'architecture par une surveillance du temps interne. |
| Actionneur | Convertisseur numérique analogique DAC | Quatre valeurs numérisées qui se situent dans un intervalle 0… 4095. | Une tension de 0 V-3,3 V. | Restitution de la valeur numérique obtenue par les contrôleurs flous MIF1 et MIF2 en quatre différents signaux analogiques identiques AO0, AO1, AO2 et AO3. |

Tableau 5-8 : Les propriétés de chaque bloc fonctionnel

### 5.4.4 Détermination des taux de défaillances

#### 5.4.4.1 Taux de défaillance de sous-système 1

Le taux de défaillance des pannes dangereuses pour le sous-système 1 sera calculé pour chaque composant du sous-système en tenant compte du facteur de couverture du test, du facteur de sécurité et du taux de défaillance de chaque composant.

Les modes de défaillance d'alimentation sont connus et peuvent être complètement détectés par un test de diagnostics afin de le classer en tant que composant de type A avec une valeur de sécurité de S = 10 %. Mais comme il n'existe pas de test de diagnostics pour la détection d'une défaillance au niveau d'alimentation, le taux de couverture de tests de diagnostics est donc de DC = 0 %.

L'alimentation se compose d'une alimentation d'une tension de 3,3 V, 2,5 V et 1,2 V d'une structure 1oo1. Le taux de défaillance non détectable, et de défaillance dangereuse détectable du composant d'alimentation est formé de la somme des taux de défaillance de chaque élément entrant dans le circuit de ce dernier.

Étant donné le taux de défaillance de base des composants d'alimentation, la couverture de diagnostic et le facteur de sécurité, le taux de défaillances dangereuses détectées et le taux de défaillances dangereuses non détectées peuvent être déduites des formules suivantes :

$$\lambda^{DU} = \lambda^{D}(1-DC) \quad , \quad \lambda^{DD} = \lambda^{D} DC \quad et \quad \lambda^{D} = \frac{\lambda}{2} S \qquad (5\text{-}1)$$

Les valeurs des taux de défaillances dangereuses d'alimentation de structure 1oo1 sont fournies par la norme Siemens 95000 [SSN 86] et représentées dans le Tableau 5-9 :

| **Taux de défaillance des composants de l'alimentation λ =5,01 E -06 ($h^{-1}$), S = 10 %, DC = 0 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse ($h^{-1}$)** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 2,50 E-07 | 2,50 E-07 | 0,00 E+00 |

Tableau 5-9 : Taux de défaillance de l'alimentation

La détection d'une défaillance dangereuse au niveau d'horloge à cause d'un dérivé de l'horloge principale à cause d'un défaut du matériel, des perturbations électrostatiques, des ondes magnétiques, etc. est réalisée par les tests de diagnostic suivants :

- la détection de la dérive de l'horloge à partir de l'observation de l'horloge avec l'aide d'un deuxième temps de base DCM ;
- le test du chien de garde.

Le taux de couverture DC pour le traitement d'horloge est de DC = 60 % et pour le chien de garde est de DC = 99 %. Les deux composants sont de type B et possèdent une valeur de sécurité de S = 50 %

Les valeurs des taux de défaillances dangereuses d'horloge de structure 1oo1 sont représentées par le Tableau 5-10 :

| **Taux de défaillance des composants de l'horloge (FPGA) λ = 3,76 E-08 ($h^{-1}$), S = 50 %, DC = 60 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse ($h^{-1}$)** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 9,40 E-09 | 3,76 E-09 | 5,64 E-09 |

Tableau 5-10 : Taux de défaillance du composant d'horloge (FPGA)

Le composant filtre passe-bas, le convertisseur analogique numérique ADC et l'amplificateur de gain programmable sont de type B et possèdent une valeur de sécurité S = 50 % et un taux de couverture de diagnostic de DC = 60 %.

Les valeurs des taux de défaillances dangereuses du convertisseur analogique numérique ADC avec le composant de filtre passe-bas et l'amplificateur de gain programmable de structure 1oo1 sont représentées sur le Tableau 5-11 :

| **Taux de défaillance des composants du convertisseur ADC λ = 5,01 E-08 [$h^{-1}$], S = 50 %, DC = 99 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse ($h^{-1}$)** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 1,25 E-08 | 1,25 E-10 | 1,24 E-08 |

Tableau 5-11 : Taux de défaillance des composants du convertisseur ADC

Le taux de défaillance dangereuse non détectable, et de défaillance dangereuse détectable du sous-système 1 est formé par l'addition des taux de défaillance de chaque bloc fonctionnel 1, 2 et 3. Compte tenu des formules mentionnées auparavant, il en résulte :

| Sous-système | $\lambda^{D}$ ($h^{-1}$) | $\lambda^{DD}$ ($h^{-1}$) | $\lambda^{DU}$ ($h^{-1}$) |
|---|---|---|---|
| Sous-système 1 | 2,72 E-07 | 1,8 E-08 | 2,54 E-07 |

Tableau 5-12 : Taux de défaillance de sous-système 1

### 5.4.4.2 Taux de défaillance du sous-système 2

Les valeurs des taux de défaillances dangereuses de chien du garde implémenté dans la puce FPGA de structure 1oo2 sont représentées par le Tableau 5-13 :

| **Taux de défaillance du chien de garde<br>λ = 1,64 E-08 ($h^{-1}$), S = 50 %, DC = 99 %, β = 2 % et $\beta_D$ = 1 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse ($h^{-1}$)** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 4,10 E-09 | 4,10 E-11 | 4,06 E-09 |

Tableau 5-13 : Taux de défaillance du chien du garde

Les modes de défaillance du moteur d'inférence floue ne sont pas connus et ne peuvent pas être complètement détectés par un test de diagnostics. Ainsi, on peut le noter de type B, avec une valeur de sécurité de S = 50 %.

Le taux de couverture du sous-système 2 appartient à la catégorie élevée, avec une valeur de sécurité de 99 %.

Les valeurs des taux de défaillances dangereuses du moteur d'inférence floue implémenté en FPGA de structure 1oo2 sont représentées dans le Tableau 5-14 :

| **Taux de défaillance du moteur d'inférence floue implémenté dans FPGA<br>λ = 3,97 E-09 par [h]<br>S = 50 % et DC = 99 %, β = 2 % et $\beta_D$ = 1 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse ($h^{-1}$)** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 9,93 E-10 | 9,93 E-12 | 9,83 E-10 |

Tableau 5-14 : Taux de défaillance du sous-système 2

#### 5.4.4.3 Taux de couverture du sous-système 3

Le convertisseur numérique analogique DAC est considéré de type B, avec une valeur de sécurité S = 50 %, comme le démontre explicitement l'analyse AMDEC. Le taux de couverture des tests de diagnostics s'élève donc à 60 %.

Les valeurs des taux de défaillances dangereuses du convertisseur numérique analogique de structure 1oo1 sont représentées sur le Tableau 5-15 :

| **Taux de défaillance du convertisseur DAC λ = 2,5 E-09 ($h^{-1}$), S = 50 % et DC = 60 %** | | |
|---|---|---|
| **Taux de défaillance dangereuse [h]** | **Taux de défaillance dangereuse non détectée ($h^{-1}$)** | **Taux de défaillance dangereuse détectée ($h^{-1}$)** |
| 6,25 E-09 | 2,5 E-09 | 3,75 E-09 |

Tableau 5-15 : Taux de défaillance du sous-système 3

### 5.4.5 Détermination de la probabilité moyenne de défaillance sur demande

L'architecture des composants d'alimentation, de l'horloge (FPGA), de convertisseur analogique-numérique et du convertisseur numérique-analogique est d'une structure 1oo1, alors la formule que l'on utilisera dans le calcul est la suivante [CEI 06] :

$$PFDavg = \lambda^{D} t_{CE} \tag{5-2}.$$

Par contre l'architecture du composant chien de garde et du MIF est d'une structure redondante 1oo2, alors la formule que l'on utilisera dans le calcul est la suivante [CEI 06] :

$$PFDavg = 2(1-\beta)\,\lambda^{DU}((1-\beta)\lambda^{DU} + (1-\beta_D)\lambda^{DD} + \lambda^{SD})t_{CE'}t_{GE'} + \beta_D\lambda^{DD}MTTR + \beta\lambda^{DU}(\frac{T_i}{2} + MTTR) \tag{5-3}.$$

La PFDavg du MIF par la méthode de bloc-diagramme de fiabilité est calculée par la combinaison de la probabilité de défaillance de tous les sous-systèmes assurant ensemble la fonction de sécurité. Elle est exprimée par les formules suivantes [CEI 00] sous l'hypothèse d'événements rares :

$$PFD_{avg} = PFD_{Capteur} + PFD_{Logique} + PFD_{Actionneur} = PFD_{Ali} + PFD_{SIF\_FPGA} + PFD_{WD\_FPGA} + PFD_{CLK\_FPGA} + PFD_{AD} + PFD_{DA} \tag{5-4}.$$

La probabilité moyenne de défaillance PFDavg est représentée sur le Tableau 5-16 pour différents temps de mission (*proof test* Ti = 1 an, 3 ans, 5 ans et 10 ans) et une durée moyenne de rétablissement de 8 heures.

| | **Ti = 1 an** | **Ti = 3 ans** | **Ti = 5 ans** | **Ti = 10 ans** |
|---|---|---|---|---|
| PFDavg | 1,15 E-03 | 3,44 E-03 | 5,73 E-03 | 1,15 E-02 |

Tableau 5-16 : Valeur de PFDavg pour différentes valeurs de Ti

La Figure 5-7 montre l'évolution de la probabilité moyenne de défaillance sur demande PFDavg au cours du temps du MIF étudié. Les composants du système sont testés aux intervalles de temps précisés précédemment. On détermine l'impact du temps de mission sur la valeur PFDavg du MIF.

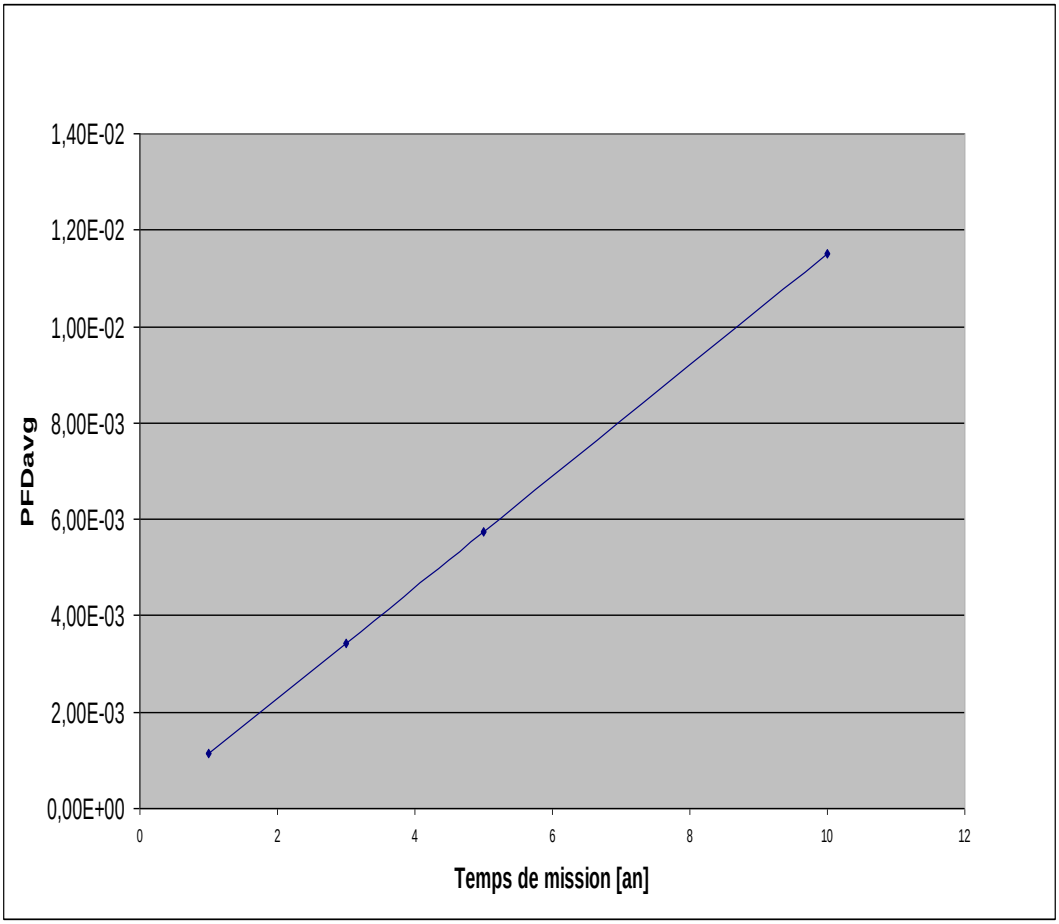



Figure 5-7 : Impact du temps de mission sur la valeur PFDavg du MIFS

La valeur PFDavg résultante pour un temps de mission Ti = 1an est de 1,15E -03. Si en retard le temps de mission Ti d'une période de 10, la valeur PFDavg passe à 1,15E -02, ce qui donne une variation du niveau de sécurité pour le MIF étudié, d'un niveau de SIL2 (PFDavg $\in$ $[10^{-4},10^{-3}]$) à un niveau SIL1 (PFDavg $\in$ $[10^{-3},10^{-2}]$).

## 5.5 Modélisation par l'arbre de défaillance

La modélisation par arbre de défaillance est l'une des méthodes les plus exploitées dans les analyses des performances des systèmes instrumentés de sécurité. Elle a pour objectif le recensement des causes entraînant une défaillance dangereuse du système [VIL 88]. La Figure 5-8 représente l'arbre de défaillances du MIF qui commence par une défaillance du MIF à cause d'un événement indésirable (sommet) et de déterminer ses causes.

Une défaillance dangereuse a affecté l'exécution de la fonction du système. Cette défaillance peut avoir été causée par le dysfonctionnement de l'alimentation sur la maquette, par une défaillance dangereuse subie par le convertisseur analogique-numérique ADC, un état de défaillance dangereuse des deux systèmes d'inférence floue mise en œuvre dans la puce FPGA, la défaillance des deux chiens de garde, ou encore de celle du convertisseur numérique-analogique DAC.

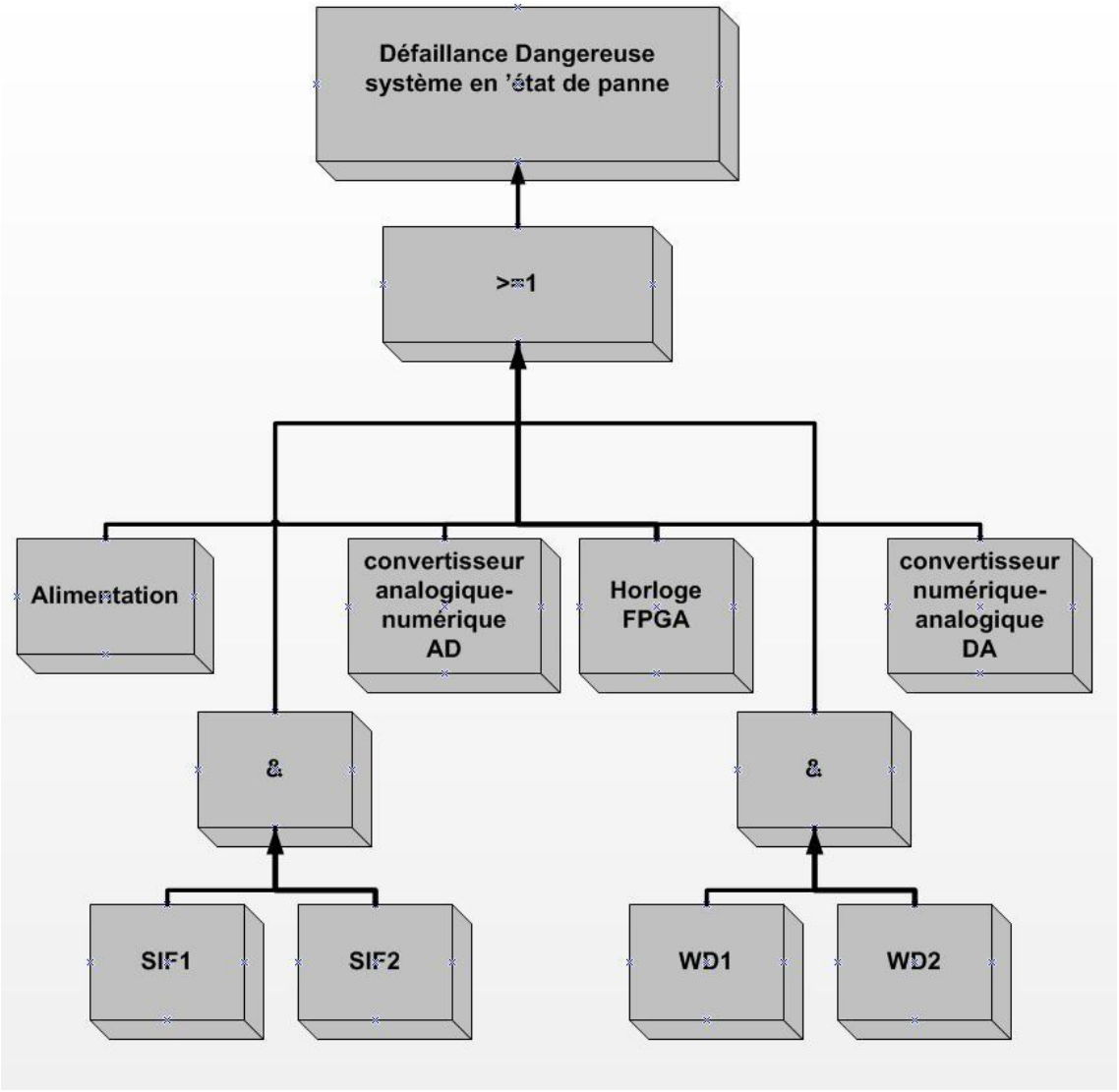



Figure 5-8 : Arbre des causes du MIFS

L'analyse par l'arbre de défaillance est effectuée en deux phases ; une qui est qualitative, où l'on détermine la fonction logique du système en terme de l'ensemble de ses défaillances minimales, et l'autre est dite quantitative par le calcul de la probabilité d'occurrence de l'événement indésirable (PFDavg).

Cette méthode de l'arbre de défaillances consiste à chercher toutes les combinaisons possibles d'événements entraînant une défaillance du système.

Dans ce cas, en utilisant la méthode proposée par [ISA 02] pour une architecture au moins 1 parmi 1 (1oo1), la probabilité moyenne de défaillance est exprimée par la formule suivante [ISA 02] :

$$PFDavg = \lambda^{DU} \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2} \tag{5-5}$$

Avec :

- $\lambda^{DU}$ taux de défaillance dangereuse non détecté ;
- $\lambda^{D}$ taux de défaillance dangereuse ;
- $T_i$ le temps d'intervalle entre les tests périodiques sur le système.

Pour une architecture 1 parmi 2 avec diagnostic (1oo2D), la probabilité moyenne de défaillance est exprimée par la formule suivante [ISA 02] :

$$\begin{aligned} PFDavg = ((1-\beta)\lambda^{DU})^2 \times \frac{T_i^2}{3} + \left[(1-\beta)\lambda^{DU}\lambda^{DD} MTTR \times T_i\right] \\ + \beta\,\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2} \end{aligned} \tag{5-6}$$

Le taux de défaillance non détecté est quantifié par le facteur β de la cause commune de défaillance qui est une valeur estimée entre 0 et 20 %. D'après des applications industrielles sur les systèmes instrumentés de sécurité (SIS) le facteur bêta est estimé dans la plage de 0,1 à 2 % [AES 99].

Pour un facteur bêta d'une valeur β = 2 % représentant respectivement la proportion de défaillances de causes communes détectées liées au taux de couverture de diagnostic DC, et à partir des taux de défaillance de chaque composant, la probabilité de défaillances sur demande par l'arbre des causes du MIF est calculée à partir des formules [ISA 02] telles que mentionnées ci-dessus, et définie de la manière suivante :

$$\begin{aligned} PFD_{SIFS}(T_i) = PFD_{ALI}(T_i) + PFD_{SIF}(T_i) + PFD_{AD}(T_i) \\ + PFD_{WD}(T_i) \quad + PFD_{CLK\_FPGA}(T_i) + PFD_{DA}(T_i) \end{aligned} \tag{5-7}$$

La probabilité de défaillance sur demande du système complet MIFS est calculée à partir de la formule suivante :

$$PFD_{SIFS}(T_i) = (\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{ALI} + (\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{AD} + (\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{DA} +$$
$$(\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{CLK\_FPGA} + (((1-\beta)\lambda^{DU})^2 \times \frac{T_i^2}{3} + \left[(1-\beta)\lambda^{DU}\lambda^{DD} MTTR \times T_i\right] + \quad (5\text{-}8)$$
$$\beta\,\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{SIF} + (((1-\beta)\lambda^{DU})^2 \times \frac{T_i^2}{3} + \left[(1-\beta)\lambda^{DU}\lambda^{DD} MTTR \times T_i\right] +$$
$$\beta\,\lambda^{DU} \times \frac{T_i}{2} + \lambda^{D} \frac{T_i}{2})_{WD}$$

La PFDavg du MIF est calculée par la combinaison de la probabilité moyenne de tous les éléments assurant ensemble la fonction de sécurité. Nous supposons ainsi que l'on ne teste pas fonctionnellement chaque sous-système indépendamment les uns des autres, mais le système complet. Les valeurs numériques des paramètres caractéristiques des composants tels que le taux de défaillance, le taux de couverture DC et le facteur de défaillance de cause commune sont représentées dans le Tableau 5-17 :

| **Composant du MIF** | **λD($h^{-1}$)** | **DC** | **β** | **MTTR (h)** |
|---|---|---|---|---|
| Alimentation | 2,5 E-07 | 0 | - | 8 |
| Horloge (FPGA) | 9,40 E-09 | 60 % | - | 8 |
| Convertisseur ADC | 1,25 E-08 | 60 % | 2 % | 8 |
| Système inférence floue | 9,93 E-10 | 99 % | 2 % | 8 |
| Chien de garde | 4,1 E-09 | 99 % | 2 % | 2 % |
| Convertisseur DAC | 6,25 E-09 | 60 % | - | 8 |

Tableau 5-17 : Données numériques

La PFDavg est présentée sur différents intervalles de temps de mission Ti (*proof test* Ti = 1 an, 3 ans, 5 ans et 10 ans) et pour une durée moyenne de rétablissement de 8 heures. Ces résultats sont regroupés dans le tableau 5-18 :

Les résultats obtenus par la méthode de l'arbre de défaillance ne sont pas similaires aux résultats obtenus par la méthode de bloc-diagramme de fiabilité, la valeur PFDavg résultante pour un temps de mission Ti = 1 an varie de 2,39E -03 jusqu'à 1,19E -02, ce qui donne une variation du niveau de sécurité pour le MIF étudié, d'un niveau de SIL2 (PFDavg $\in$ [$10^{-4}$,$10^{-3}$]) à un niveau de SIL1 dans un temps de mission de 5 ans au lieu de 10 ans obtenus par la méthode de bloc-diagramme de fiabilité. Cela est dû à la méthode qui ne tient pas compte dans ses formules des valeurs numériques des paramètres caractéristiques des composants, le facteur de défaillance de cause commune non détecté $\beta_D$ ainsi que le taux de couverture DC.

| | **Ti = 1 an** | **Ti = 3 ans** | **Ti = 5 ans** | **Ti = 10 ans** |
|---|---|---|---|---|
| **PFDavg** | 2,39 E-03 | 7,17 E-03 | 1,19 E-02 | 2,39 E-02 |

Tableau 5-18 : Valeur de PFDavg pour différentes valeurs de Ti

Cela nous amène dans le cadre de l'évaluation de la performance du MIFS à vérifier l'effet de l'imprécision du taux de couverture des composants sur la valeur de la probabilité moyenne de défaillance sur demande. Pour tenir compte de l'imprécision, la valeur de couverture DC peut être représentée par un intervalle.

$DC \in [DC_{min}, DC_{max}]$

De ce fait, les différents taux de défaillances dangereuses deviennent :

$$\lambda_{\max}^{DU} = \frac{\lambda}{2}(1 - DC_{\min}) \quad , \quad \lambda_{\min}^{DD} = \lambda_{\min}^{SD} = \frac{\lambda}{2} DC_{\min} \quad ; \quad \lambda_{\min}^{D} = \frac{\lambda_{\min}}{2}$$

$$\lambda_{\min}^{DU} = \frac{\lambda}{2}(1 - DC_{\max}) \quad , \quad \lambda_{\max}^{DD} = \lambda_{\max}^{SD} = \frac{\lambda}{2} DC_{\max} \quad ; \quad \lambda_{\max}^{D} = \frac{\lambda_{\max}}{2} \qquad (5\text{-}9)$$

Les probabilités moyennes de défaillance sur demande PFDavg pour les différentes valeurs de DC corrigées. Dcmin = DC -30, DC -20 et Dcmax = DC sont représentées dans le Tableau 5-19 :

| **Méthode de calcul** | **La valeur PFDavg par [an]** | | |
|---|---|---|---|
| | **$\beta$ = 2 % $\beta_D$ = 1 % Ti = 8 760 MTTR = 8 h** | | |
| | DC-30 | DC-20 | DC |
| **Bloc-diagramme de fiabilité** | 1,19 E-02 | 0,17 E-03 | 1,15 E-03 |
| **Arbre de cause** | 2,43 E-02 | 2,41 E-03 | 2,39 E-03 |

Tableau 5-19 : Imprécision de la valeur de couverture DC sur la valeur PFDavg

Selon le Tableau 5-19, la probabilité moyenne de défaillance sur demande obtenue par l'arbre de cause en tenant compte de l'imprécision du facteur de couverture de diagnostic est comprise entre 2,39E -03 et 2,43E -02, ce qui classe le régulateur MIFS par un niveau de sécurité SIL2. La probabilité moyenne de défaillance sur demande par bloc-diagramme de fiabilité en tenant compte de l'imprécision du facteur de couverture diagnostic est comprise entre 1,15E -03 et 1,19E -02, ce qui classe le régulateur MIFS par niveau de sécurité SIL1.

Si l'imprécision du facteur de couverture de diagnostic n'est pas prise en considération, le niveau de sécurité du régulateur MIFS sera de SIL2 au lieu de SIL1.

Le calcul de la probabilité moyenne de défaillance sur demande PFDavg est effectué par notre propre calcul. La détermination de la valeur PFDavg a été calculée à partir de rétablissement de taux de pannes de base de chaque composant du MIF, le taux de couverture DC associé, le modèle du facteur β, le temps de mission Ti, la durée moyenne de MTTR et les formules probabilistes déjà citées dans les paragraphes précédents.

## 5.6 Modélisation par graphe de Markov

Le moteur d'inférence floue étudié est composé de trois sous-systèmes :

- La partie capteur au niveau de la carte en architecture 1oo1, constituée d'un convertisseur analogique-numérique qui a acquis les valeurs analogiques des deux canaux d'entrée AI0 et AI1.
- La partie logique (MIF) implémentée en FPGA en architecture 1oo2D.
- La partie actionneur en architecture 1oo1, composée par un convertisseur numérique-analogique avec quatre canaux de sortie.

Son bloc-diagramme de fiabilité est fourni à la Figure 5-9.

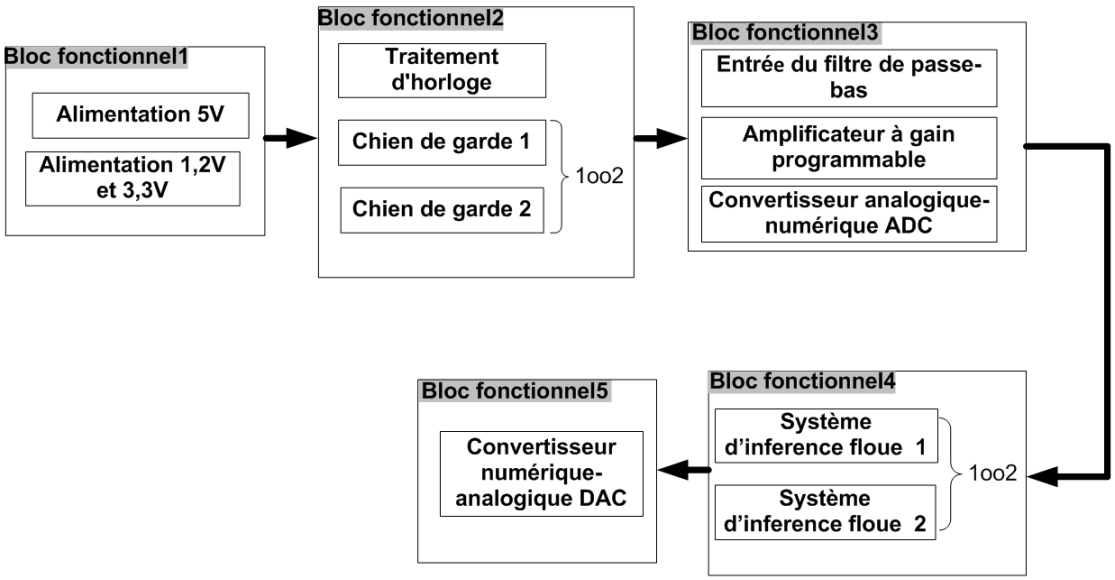



Figure 5-9 : Le bloc fonctionnel du moteur d'inférence floue

Notre objectif est de déterminer la valeur du MIF, à partir des paramètres caractéristiques tels que le taux de couverture de diagnostic et le facteur de défaillance de cause commune, en utilisant les chaînes de Markov [MSB 08].

Le moteur d'inférence floue se compose de 10 composants, chacun pouvant avoir deux états : opérant ou en défaillance dangereuse détectée et non détectée.

La réalisation de la chaîne de Markov par l'espace d'état consiste en l'analyse de $3^{10}$ états possibles. Afin de simplifier, nous proposons d'évaluer la PFDavg de chacun des sous-systèmes en série qui constituent le MIF et ensuite les additionner pour obtenir la PFDavg globale.

La PFDavg du MIF est calculée par la combinaison de la probabilité moyenne de tous les éléments assurant ensemble la fonction de sécurité. Nous supposons ainsi que l'on ne teste pas le fonctionnement de chaque sous-système indépendamment les uns des

autres, mais le système complet. La PFDavg du MIF est exprimée par les formules suivantes [CEI 00] sous l'hypothèse d'événements rares :

$$\begin{aligned} PFD_{avg} &= PFD_{Sens} + PFD_{Logique} + PFD_{Act} \\ &= PFD_{Ali} + PFD_{SIF\_FPGA} + PFD_{WD\_FPGA} + PFD_{CLK\_FPGA} + PFD_{AD} + PFD_{DA} \end{aligned} \quad (5\text{-}10)$$

La probabilité de défaillance sur demande du système MIFS est calculée à partir de la démarche proposée par [GBL 11] et [MSB 10]. Elle est égale à l'indisponibilité moyenne calculée sur la durée de mission Ti ou éventuellement sur l'intervalle de test [0, Ti] si tous les composants sont testés simultanément. Elle est exprimée par la formule suivante [MSB 10] :

$$PFDavg = \frac{1}{k \cdot \Delta t} \cdot \sum_{n=0}^{k} \sum_{Sj} p^{(n)}(S_j) \cdot \lambda_{ij} \cdot \Delta t \quad (5\text{-}11)$$

Où k.Δt ∈ [0,Ti].

Ti est le temps de mission, Si sont les états de défaillances dangereuses et $p^n(S_j)$ est la probabilité d'être dans un de ces états à l'instant n.

L'équation 5-11 représente la probabilité que le système étudié soit dans l'état Sj à l'instant n à partir de n'importe quel autre état Si à l'instant (n-1) selon une probabilité de transition λij de Si vers Sj définie dans la matrice de transition P = (λij).

La probabilité de défaillance du MIFS lors de sa sollicitation est déterminée à partir des paramètres caractéristiques de ses composants. Les valeurs numériques des paramètres caractéristiques des composants du MIFS sont données dans le Tableau 5-20 :

| Composant du MIF | Architecture | λD($h^{-1}$) | DC | β | $\beta_D$ | MTTR (h) |
|---|---|---|---|---|---|---|
| Alimentation | 1oo1 | 00 E+00 | 0 | - | - | 8 |
| Horloge (FPGA) | 1oo1 | 5,64 E-09 | 60 % | - | - | 8 |
| Convertisseur ADC | 1oo1 | 7,52 E-09 | 60 % | - | - | 8 |
| Convertisseur numérique analogique DAC | 1oo1 | 7,50 E-09 | 60 % | - | - | 8 |
| Système inférence floue | 1oo2 | 1,01 E-07 | 99 % | 2 % | 1 % | 8 |
| Chien de garde | 1oo2 | 4,06 E-09 | 99 % | 2 % | 1 % | 8 |

Tableau 5-20 : Données numériques

L'unité capteur en architecture 1oo1 est composée d'alimentation, d'horloge de FPGA et de convertisseur analogique-numérique et ses composants. Le comportement de ce sous-système est périodiquement testé (intervalle entre tests égal à Ti), peut être

schématisé par un modèle markovien multiphases [MSB 08] comme sur la Figure 5-10 :

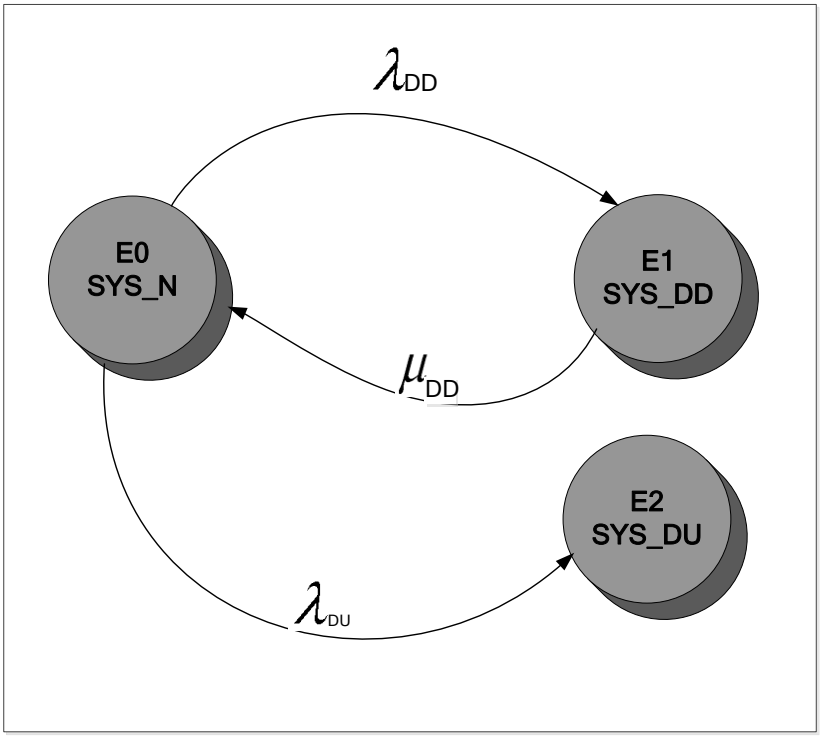



Figure 5-10 : Modèle de Markov des unités d'architecture un parmi un 1oo1

La figure modélise le système MIFS. Les trois états décrits par le modèle markovien et les phases s'enchaînent au moment du test de la manière suivante :

- Le système est dans l'état E0, il est complètement opérationnel et n'existe pas de défaillance. Dans cet état la fonction de sécurité est validée (le système peut répondre à une sollicitation). De cet état, le système peut basculer vers deux autres états.
- Le système est dans l'état E1, la fonction de sécurité du système est valide. Le système passe à cet état après une défaillance dangereuse détectée puis réparée. Avec $\mu_{DD}$ qui représente le taux de transition spécifique aux défaillances dangereuses détectées par le test de diagnostics.
- Le système est dans l'état E2, la fonction de sécurité n'est plus réalisée, un ou plusieurs composants étant défaillants à cause d'une défaillance dangereuse non détectée.

La matrice de transition sur l'architecture du MIFS est obtenue de la manière suivante :

$$P = \begin{bmatrix} P_{00} P_{01} P_{02} \\ P_{10} P_{11} P_{12} \\ P_{20} P_{21} P_{22} \end{bmatrix} = \begin{bmatrix} 1-\lambda_{00} & \lambda_{01} & \lambda_{02} \\ \lambda_{10} & 1-\lambda_{11} & \lambda_{12} \\ \lambda_{20} & \lambda_{21} & 1-\lambda_{22} \end{bmatrix}$$

La valeur du taux de défaillances détectée et non détectée du capteur est calculée par l'addition des taux de défaillance de tous les composants du sous-système de l'unité capteur (alimentation, horloge de FPGA et le convertisseur analogique-numérique ADC). Elle est donnée dans le Tableau 5-21 :

| Composant du MIFS | Architecture | λDU($h^{-1}$) | λDD($h^{-1}$) | DC | Ti (an) | MTTR (h) |
|---|---|---|---|---|---|---|
| Capteur | 1oo1 | 2,59E-09 | 1,32E -09 | 60 % | 1 | 8 |

Tableau 5-21 : Données numériques du capteur

Après l'insertion des valeurs de transition dans la P-Matrice, on obtient la matrice de transition du MIFS suivante :

$$P = \begin{bmatrix} 9,99E-01 & 1,32E-09 & 2,59E-09 \\ 0,125 & 0,875 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Maintenant que la matrice de transition a été formée, la probabilité de défaillance sur demande du système MIFS par la chaîne de Markov pour l'unité du capteur est calculée à partir de la démarche proposée par Goble [GBL 10] :

À partir de l'état de départ spécifié par la matrice S = (1, 0, 0), ainsi que de la matrice de la transition P du MIFS on peut calculer la matrice $S_i$ et la valeur PFD associée pour n'importe quel temps d'itérations en multipliant $S_{i-1}$ par la matrice de transition P.

La probabilité de défaillance sur demande du système MIFS est calculée à partir de la démarche proposée par [GBL 10] :

$$S^i = S^{i-1} \times P \tag{5-12}$$

En considérant que le système est toujours en service à partir de l'état E0. La probabilité de départ S est définie comme suit :

$$S^i = [1\ 0\ 0]$$

La matrice Sn du MIFS pour n'importe quel intervalle de temps particulier est obtenue en multipliant Sn-1 par la matrice de transition.

$$S^1 = S^0 \times P = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 9,9910^{-01} & 1,3210^{-09} & 2,5910^{-09} \\ 0,125 & 0,875 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$S^2 = S^1 \times P$

...

...

$S^n = S^{n-1} \times P$

Le MIFS est hors service dans l'état E3 à cause d'une défaillance dangereuse, alors pour obtenir la probabilité moyenne de défaillance sur demande PFDavg, il faut calculer pour chaque itération la valeur PFDavgi= PFDavgi + PFDavgi(3) et la diviser par le nombre des itérations effectuées.

Pour différents nombres d'itérations équivalents à la valeur de différents temps de mission du système Ti, on obtient les valeurs PFDavg pour l'unité de capteur :

| **Ti [ans]** | **1** | **3** | **5** | **10** |
|---|---|---|---|---|
| **PFDavg** | 1,13 E-3 | 3,4 E-3 | 5,65 E-3 | 1,1 E-2 |

Tableau 5-22 : Les valeurs PFDavg pour l'unité de capteur

Le sous-système traitement a une architecture 1oo2. Son modèle markovien tenant compte à la fois du comportement normal sans et avec les défaillances de cause commune est représenté par la Figure 5-11 :

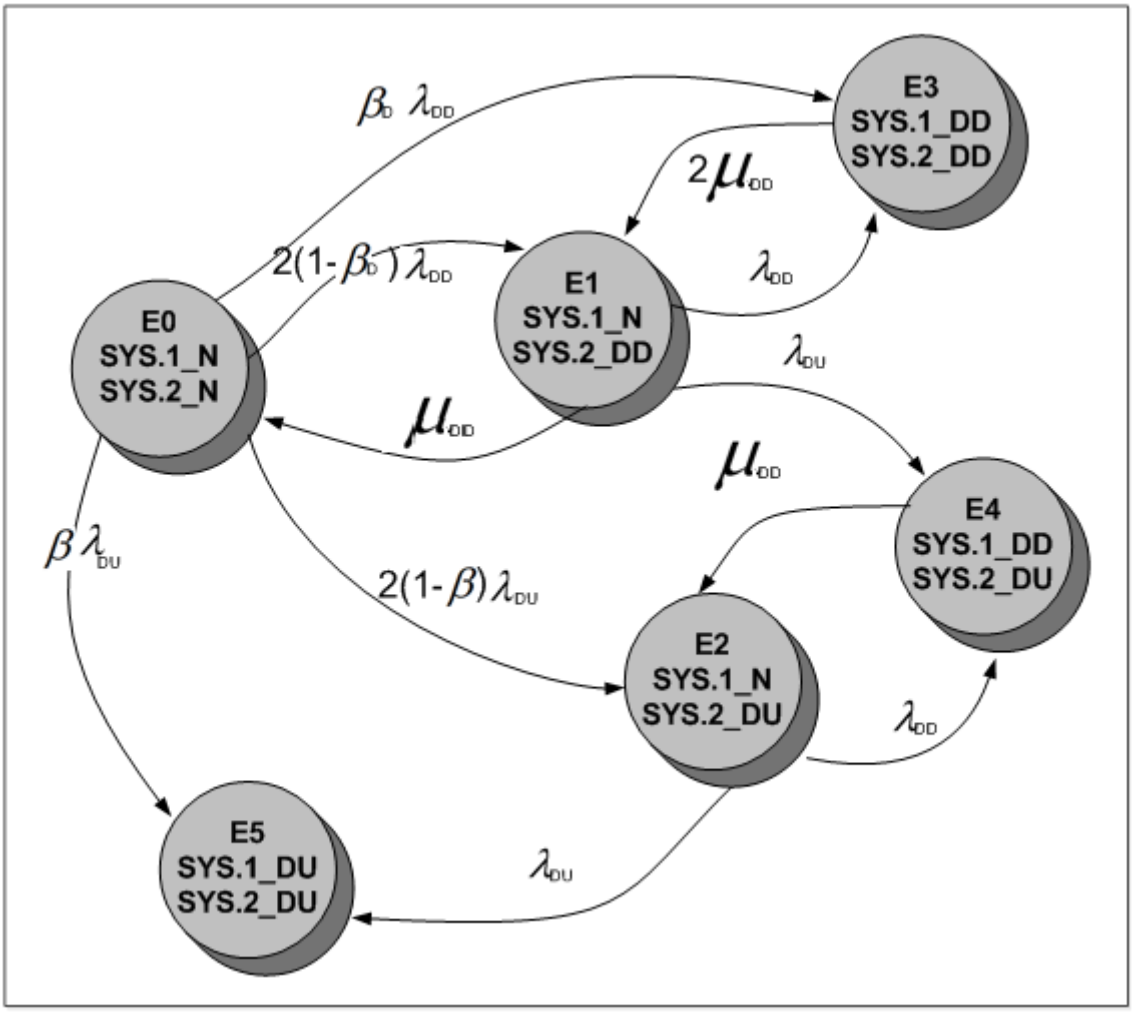



Figure 5-11: Modèle de Markov des unités d'architecture un parmi un 1oo2

La valeur du taux de défaillances détectée et non détectée et du contrôleur flou est calculée par l'addition des taux de défaillance du chien de garde et du contrôleur flou. Elle est donnée par le Tableau 5-23 :

| Composant du MIFS | Architecture | λDU($h^{-1}$) | λDD($h^{-1}$) | DC | β | $\beta_D$ | MTTR (h) |
|---|---|---|---|---|---|---|---|
| MIFS | 1oo2 | 4,06 E-09 | 4,1 E-11 | 99 % | 2 % | 1 % | 8 |

Tableau 5-23 : Données numériques du capteur

La probabilité de défaillance sur demande du système MIFS est calculée à partir de la démarche proposée par Goble [GBL 10] :

$$S^i = S^{i-1} \times P$$

En considérant que le système est toujours en service à partir de l'état E0. La probabilité de départ S est définie comme suit :

$$S^i = [1\,0\,0\,0\,0\,0]$$

La matrice Si du MIFS pour n'importe quel intervalle de temps particulier est obtenue en multipliant Sn-1 par la matrice de transition.

$$S^1 = S^0 \times P = [1\,0\,0\,0\,0\,0] \times \begin{bmatrix} 9,9910^{-01} & 1,5910^{-09} & 1,9510^{-11} & 9,8310^{-12} & 0 & 1,9910^{-13} \\ 0,125 & 0,8749 & 0 & 0 & 9,9310^{-12} & 0 \\ 0 & 0 & 9,9910^{-1} & 0 & 9,8310^{-10} & 9,9310^{-12} \\ 0 & 0,25 & 0 & 0,75 & 0 & 0 \\ 0 & 0 & 0,125 & 0 & 0,875 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$S^2 = S^1 \times P$

...

...

$S^n = S^{n-1} \times P$

Pour différents nombres d'itérations équivalents à la valeur des différents temps de mission du système Ti, on obtient les valeurs PFDavg pour l'unité de traitement :

| **Ti [ans]** | **1 an** | **3 ans** | **5 ans** | **10 ans** |
|---|---|---|---|---|
| **PFDavg** | 8,719 E-10 | 2,615 E-09 | 4,358 E-9 | 8,719 E-09 |

Tableau 5-24 : les valeurs PFDavg pour l'unité de traitement

L'unité d'actionneur en architecture 1oo1 est composée de convertisseurs numériques-analogiques. Le comportement de ce sous-système est périodiquement testé (intervalle entre tests égal à Ti), peut être schématisé par un modèle markovien multiphases (fFigure 5-12) :

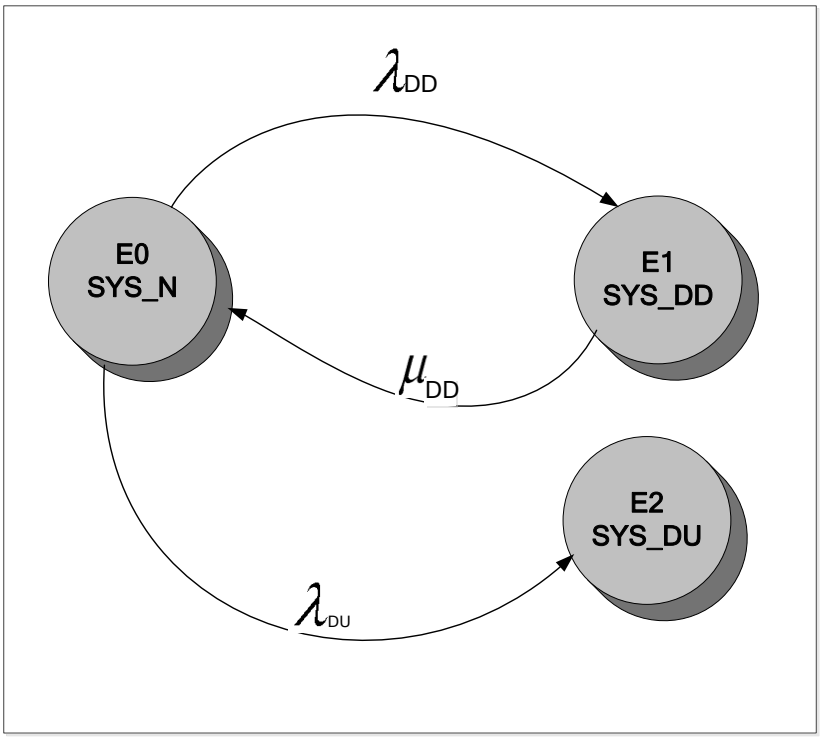



Figure 5-12 : Modèle de Markov d'unité d'actionneur de structure 1oo1

La valeur du taux de défaillances détectée et non détectée est donnée par le Tableau 5-25 :

| Composant du MIFS | Architecture | λDU(h-1) | λDD(h-1) | DC | MTTR (h) |
|---|---|---|---|---|---|
| Logic Solver | 1oo1 | 3,75 E-09 | 2,5 E-9 | 60 % | 8 |

Tableau 5-25 : Données numériques de l'actionneur

Après l'insertion des valeurs de transition dans la P-Matrice, on obtient la matrice de transition du MIFS suivante :

$$P = \begin{bmatrix} 9,9910^{-01} & 3,7510^{-09} & 2,510^{-09} \\ 0,125 & 0,875 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

En appliquant la démarche proposée par [GBL 11] pour différents nombres d'itérations équivalents à la valeur de différents temps de mission du système Ti, on obtient les valeurs PFDavg pour l'unité d'actionneur :

| Ti [ans] | 1 an | 3 ans | 5 ans | 10 ans |
|---|---|---|---|---|
| **PFDavg** | 1,1 E-3 | 3,2 E-3 | 5,48 E-3 | 1,09 E-2 |

Tableau 5-26 : les valeurs PFDavg pour l'unité d'actionneur

La PFDavg du MIFS est calculée par l'addition de la probabilité moyenne de l'unité capteur, de l'unité de traitement et de l'unité actionneur, on obtient les valeurs PFDavg du MIFS :

| Ti [ans] | 1 an | 3 ans | 5 ans | 10 ans |
|---|---|---|---|---|
| **PFDavg** | 2,23 E-03 | 6,6 E-03 | 1,11 E-03 | 2,19 E-02 |

Tableau 5-27 : les valeurs PFDavg du régulateur MIFS par la méthode Markov

## 5.7 Mise en œuvre dans le circuit FPGA

La complexité de la technologie du circuit FPGA ainsi que du moteur d'inférence floue d'architecture un parmi deux avec diagnostic (1oo2D) à implémenter nécessite une maîtrise des étapes de conception par un modèle de conception, afin de maîtriser les différentes erreurs qui peuvent se produire pendant les étapes de conception.

Le flot de conception du régulateur flou MIFS est basé sur le modèle du cycle V qui est un modèle de gestion de projet permettant, en cas d'anomalie, de limiter un retour aux étapes précédentes [BBO 81].

1. Dans la première partie, la fonctionnalité du circuit à implémenter en FPGA ainsi que l'exigence de la sécurité sont spécifiées (réalisation de la description matérielle du régulateur SIF et l'évaluation qualitative de ce dernier).
2. À ce stade une conception architecturale du système implémenté en FPGA doit être réalisée. Aussi la spécification des tests d'intégrations pour le circuit FPGA doit être créée (réalisation de la modélisation quantitative du système d'inférence floue).
3. Dans la troisième partie, les exigences relatives aux modules (V) HDL sont spécifiées.
4. À ce stade, les modules (V) HDL sont décrits d'une manière plus détaillée, à savoir, la description de l'interface et l'implémentation des fonctions.
5. Dans cette phase, le code final (V) HDL est disponible. La phase de développement est terminée et commence le processus de la vérification et de la validation.

6. Les fonctions implémentées dans le circuit FPGA seront testées par des bancs d'essai (*testbench*). La vérification est réalisée par l'examen du résultat des *testbenchs* par rapport à la spécification déjà faite dans la phase 2.
7. Dans cette phase, la validation de la mise en œuvre complète des exigences du module (V) HDL est réalisée par un vérificateur.
8. Dans cette phase, l'implémentation du code (V) HDL en FPGA est vérifiée par des *testbenchs*. La vérification est réalisée par l'examen du résultat des *testbenchs* par rapport à la spécification des tests d'intégrations déjà faite dans la phase 2.
9. Dans la phase finale, la vérification du système implémenté en FPGA au niveau des exigences sécuritaires (réalisation d'une structure redondante du régulateur MIFS, réalisation du chien de garde d'une structure redondante).

Le modèle en cycle V autorise le passage dans la partie descendante (à gauche) d'une étape à l'autre seulement lorsque l'examen (vérification) des documents créés dans l'étape actuelle est effectué par une tierce personne indépendante.

Les phases de la partie droite (Figure 5-13) doivent donner des informations sur les phases de la partie (gauche) descendante lorsque des défauts sont détectés, afin d'améliorer l'application.

La mise en œuvre et la création du banc d'essai du MIFS sur la carte Spartan-3E FPGA Starter Kit Board sont effectuées par le logiciel ISE Web de la société Xilinx. La méthode de modèle en V est utilisée pour le développement du code VHDL.

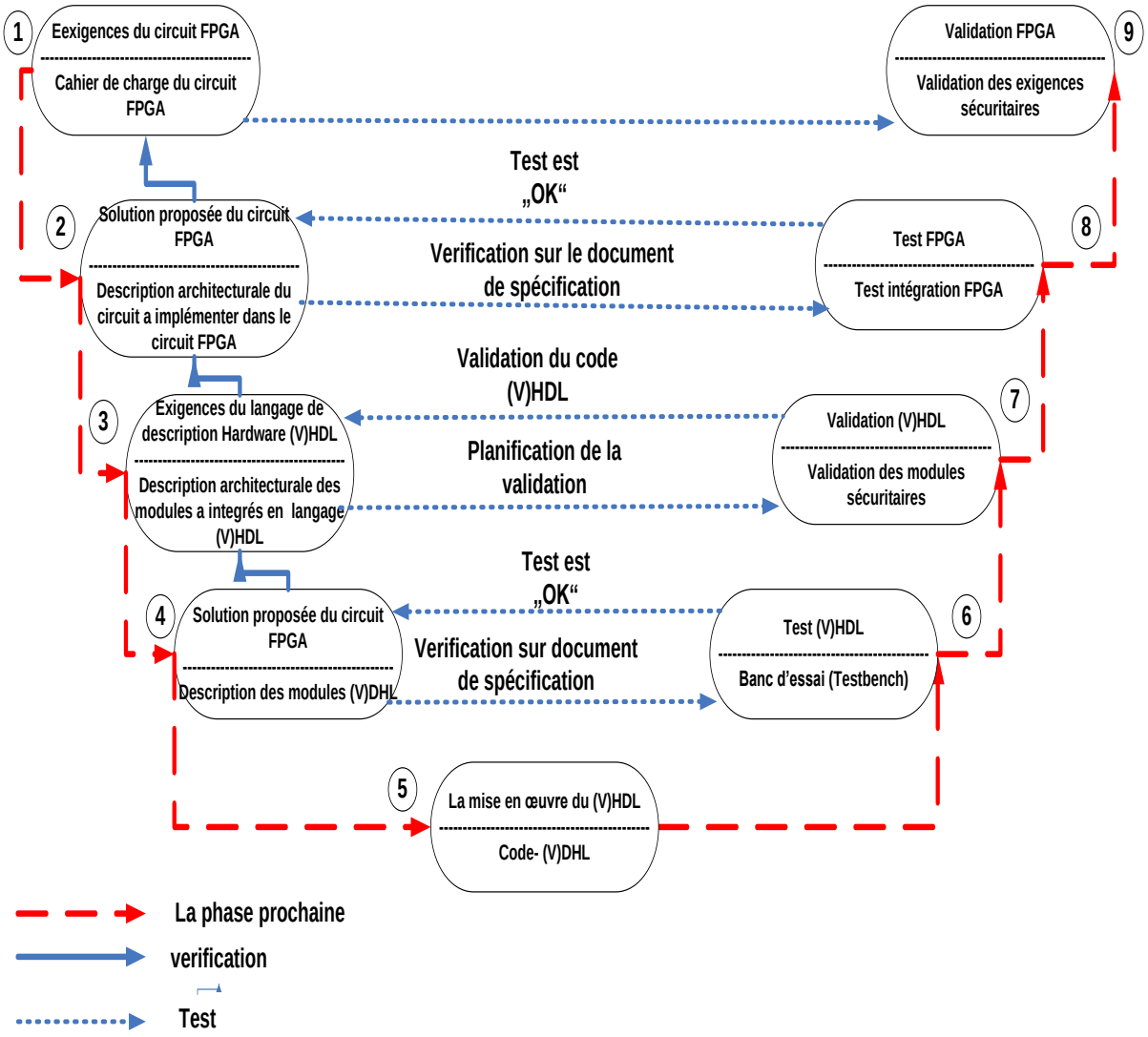



Figure 5-13 : le modèle de cycle V pour la conception du régulateur flou MIFS

Les résultats de la simulation ont satisfait la condition souhaitée d'un moteur d'inférence floue sûr de genre 1oo2D. Les figures (Figure 5-14 et Figure 5-15) représentent la simulation par banc d'essai du SIF de structure simple et avec une redondance.

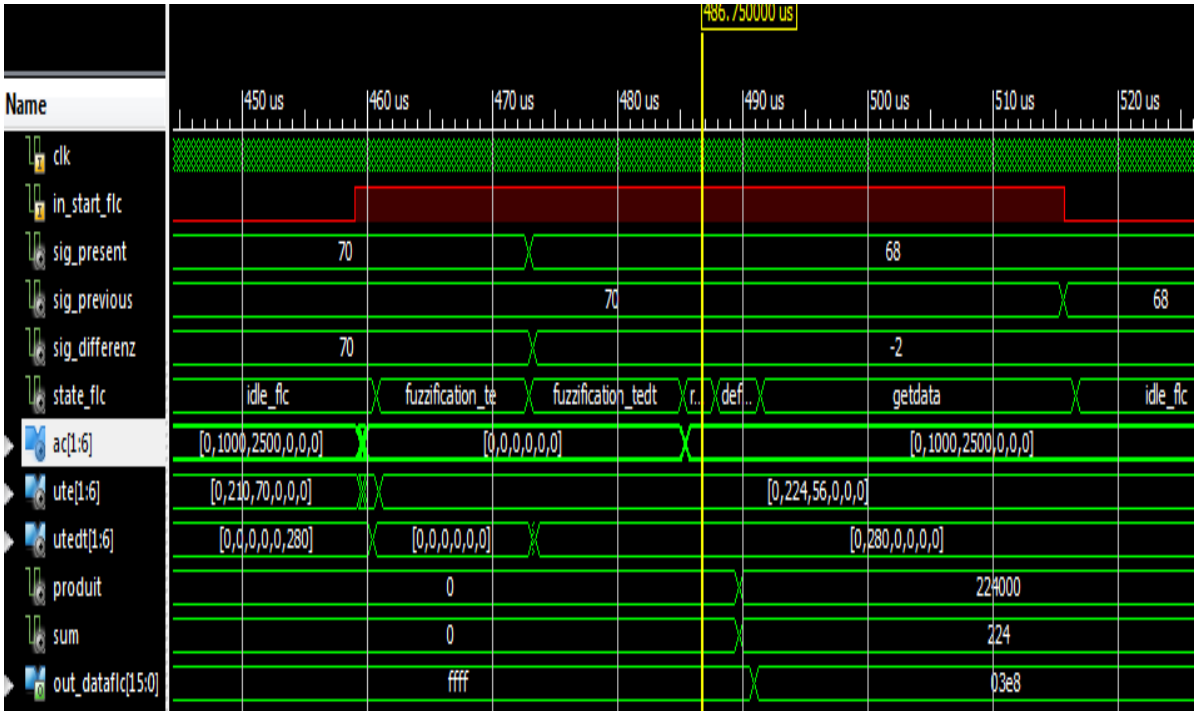



Figure 5-14 : Le résultat du régulateur SIF d'une structure Un parmi Un 1oo1

Une fois qu'il est certain que le convertisseur flou d'une structure simple 1oo1 a été correctement simulé par les bancs d'essai, un *testbench* est maintenant généré pour le moteur MIFS d'architecture 1oo2D. La Figure 5-15 montre les résultats de la simulation :

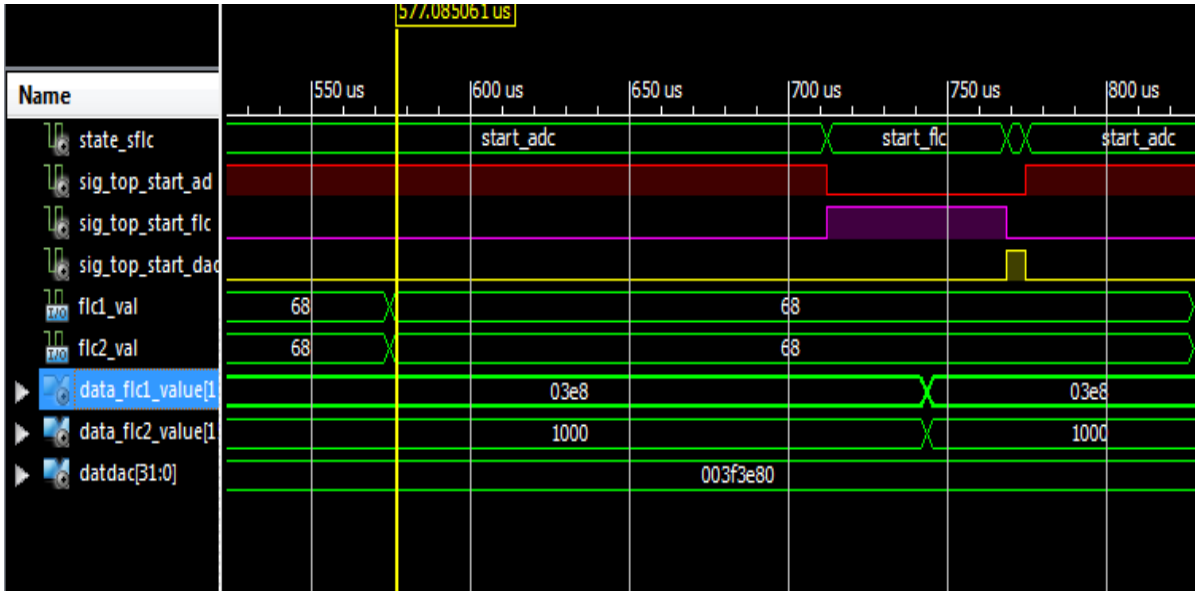



Figure 5-15 : Le moteur du MIFS d'une structure redondante 1oo2D

À chaque front montant du signal « *sig_top_start_ad* », le convertisseur analogique numérique interprète ce qui arrive sur ses pins analogiques et les retranscrit sous forme d'un nombre entier avant de le transmettre avec une période de T = 200 us ; aux

deux contrôleurs flous qui à leur tour traitent les données dans une durée de traitement de T = 57 us.

Il suit la phase de mise en œuvre du code VHDL dans le FPGA. La synthèse est faite en technologie FPGA. Dans le cadre de ce mémoire, nous avons utilisé le PFGA de la famille Xilinx (XC3S500E) qui a une capacité d'environ 500 gates et 324 portes d'entrée et sortie.

Pour la mise en œuvre en FPGA, on a commencé par le point de départ en écrivant le moteur flou MIFS par le langage VHDL. Ensuite on a utilisé Synopsis pour la synthèse en ciblant le circuit FPGA (XC3S500E de la société Xilinx). Cette étape de routage et de placement nous a permis de convertir la description VHDL en format XNF par le logiciel Web pack comme représenté par la figure suivante :

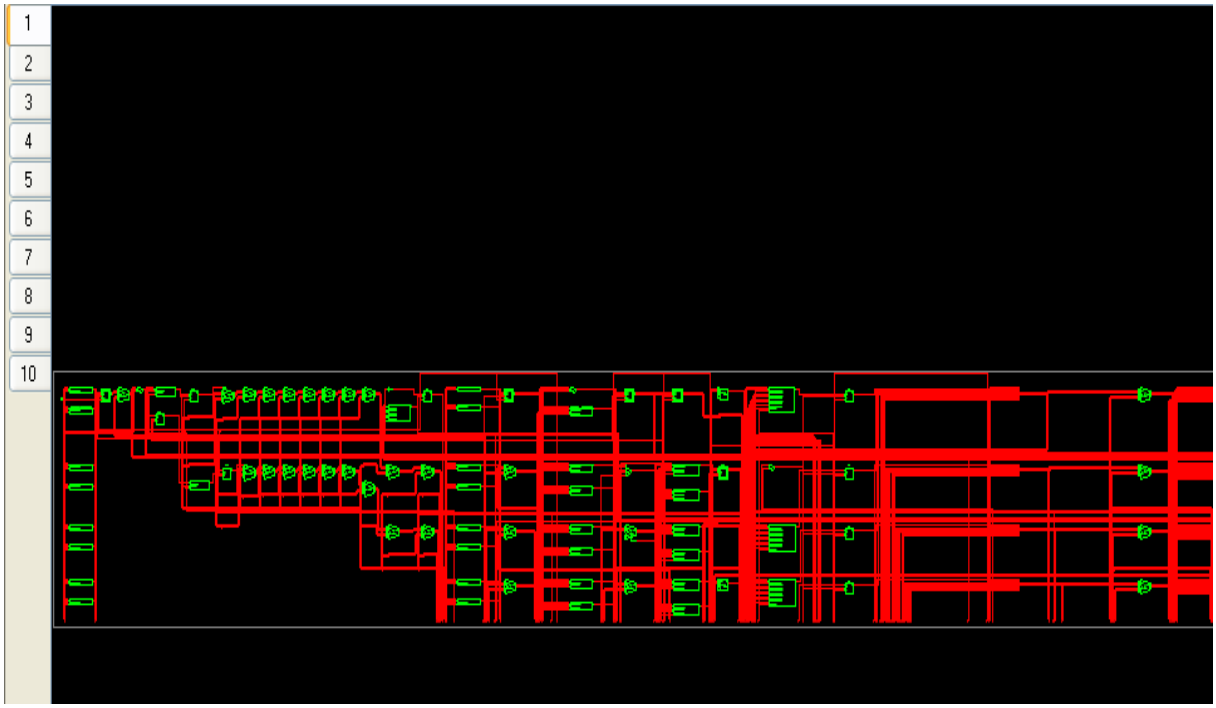



Figure 5-16 : Une partie du circuit utilisé dans la puce FPGA

Ce fichier est ensuite utilisé par Design Manager de Web pack pour fournir un fichier format BIT. C'est ce dernier qui va servir à la configuration du FPGA avec le moteur flou MIFS. Les résultats de synthèse après placement et routage par Design Manager de Xilinx utilisant le FPGA XC3S500E sont résumés dans le Tableau 5-28 :

| La famille FPGA | Slices utilisés | Bascules utilisées | Tableaux LUT | IOB utilisés | Nombre d'horloges | Multiplexeur utilisé |
|---|---|---|---|---|---|---|
| Spartan-3E XC3S500E | 1942 (41 %) | 1221 (13 %) | 3551 (38 %) | 60 (25 %) | 2 (4 %) | 15 (75 %) |

Tableau 5-28 : Résultats de synthèse du régulateur MIFS sur FPGA XC3S500E

Le Tableau 5-28 indique par quels éléments, le code VHDL du MIFS-1oo2D a été mis en œuvre dans le FPGA, soit 41 % des slices, 13 % des bascules (slices flip flops), 38 % des tableaux LUTS avec 4 entrées, 25 % des entrées et des sorties de la puce, 75 % des multiplexeurs de genre MULT18X18SIOs et 4 % de ressource de l'horloge sont utilisés, soit 2 fois plus de ressources qu'un MIF d'une structure Simple 1oo1.

Ce qui signifie que la réalisation de la fonction de sécurité par une architecture redondante est très coûteuse en termes de ressources et en coût.

L'étape finale, avant la génération du fichier de configuration, consiste en la consigne des broches. En effet, sans brochage, les ports d'entrée et de sortie du code VHDL du moteur d'inférence floue sûr de genre 1oo2D sont distribués sur des broches indéfinies du FPGA.

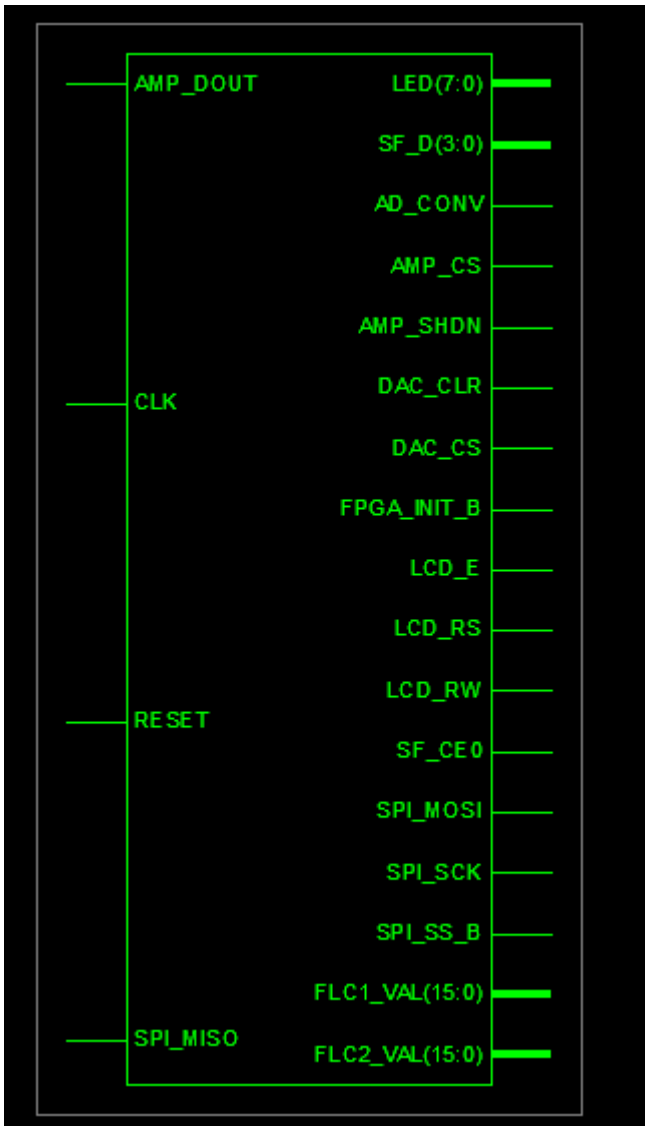



Figure 5-17 : Les entrées sorties du MIFS de structure redondante

## 5.8 Conclusion

Les résultats de calcul de la valeur de la probabilité moyenne de défaillance sur demande du moteur d'inférence floue sûr d'architecture (1oo2D), par la méthode des blocs-diagrammes, par la méthode de l'arbre de cause et par le graphe de Markov montrent que chaque méthode a ses caractéristiques d'analyse et de quantification de la fonction de sécurité.

On perçoit que la méthode des blocs-diagrammes de fiabilité (BDF) modélise le système par des blocs de diagramme fonctionnel et permet une vue d'architecture système. Par contre, la méthode de l'arbre des causes exige en plus de l'analyse fonctionnelle, la détermination des défaillances dangereuses et les événements qui peuvent être associés et qui causent la perte de la fonction de sécurité. Les résultats des trois méthodes sont presque similaires si on considère que le facteur β et le taux de couverture DC sont précis.

La méthode des graphes de Markov est la plus compliquée des trois et permet une analyse sur une longue période de façon à prédire le comportement futur du MIFS connaissant l'état présent du processus.

La récapitulation des résultats pour la probabilité moyenne d'une défaillance sur demande PFDavg du MIF est représentée par le Tableau 5-29 :

| **Système** | | **La valeur PFDavg par [an]** $\beta$ **= 2 %** $\beta_D$ **= 1 % Ti = 8 760 MTTR = 8 h** | | |
|---|---|---|---|---|
| | **Ti** | **Diagramme de fiabilité** | **Arbre de cause** | **Modèle de Markov** |
| **MIF** | **1 an** | 1,15 E-03 | 2,39 E-03 | 2,23 E-03 |
| | **3 ans** | 3,44 E-03 | 7,17 E-03 | 6,6 E-03 |
| | **5 ans** | 5,73 E-03 | 1,19 E-03 | 1,11 E-03 |
| | **10 ans** | 1,15 E-02 | 2,39 E-02 | 2,19 E-02 |

Tableau 5-29 : Résumé des résultats de la modélisation du régulateur MIFS

Dans ce chapitre, la mise en œuvre sur la puce FPGA et sa vérification ont aussi été présentées.

# Conclusion et perspectives

# Conclusion et perspectives

Les travaux que nous venons de présenter s'inscrivent dans le contexte des systèmes embarqués de contrôle-commande dédiés à la sécurité avec un contrôleur flou basé sur un moteur d'inférence floue utilisant la technologie FPGA avec une structure redondante au moins un parmi deux avec diagnostic (1oo2D) en temps réel. Nous avons proposé et validé un moteur d'inférence floue avec sécurité pour la mise en œuvre matérielle sur le circuit FPGA XCS500E de la famille Spartan-3E. Nos conclusions s'attachent aux trois parties suivantes :

- La conception du moteur d'inférence flou sûr MIFS.
- La qualification par architecture et la quantification par le calcul de la probabilité de défaillance moyenne sur demande PFDavg du MIIFS.
- La vérification et la validation de la mise en œuvre du MIFS.

Les systèmes d'inférence floue (SIF) sont basés essentiellement sur le langage C, considéré comme le langage de description logiciel et matériel. Selon le compilateur utilisé, il peut posséder des instructions capables de réaliser des comportements parallèles qui seront implémentés dans des blocs matériels avec un exécutif matériel toujours pas optimal. Un facteur qui freine la diffusion plus large de ces langages réside dans le fait que les développeurs ont besoin de connaissances matérielles, et restent ainsi difficilement maîtrisables par des non-spécialistes des systèmes embarqués. Notre approche propose d'embarquer des systèmes d'inférence floue répondant aux contraintes de sûreté de fonctionnement au niveau même du circuit FPGA à base de mémoire statique SRAM, grâce à l'utilisation de deux contrôleurs redondants. L'utilisation des circuits FPGA pour les systèmes d'inférence floue se révèle très bénéfique. En effet, les caractéristiques principales d'un moteur d'inférence floue peuvent être optimisées sans pour autant se soucier d'influencer un autre paramètre. En ce qui concerne la partie matérielle, la modification du circuit n'implique pas des dépenses dans un matériel supplémentaire, car il y aura seulement une exploitation des ressources internes dans la puce FPGA. Ajouter à cela, les autres atouts du circuit FPGA, en particulier la possibilité de mise à jour ou de correction du circuit sans pour autant que cela soit compromettant pour la chaîne de fabrication ni le coût de revient du MIF.

Sur la carte Spartan-3E FPGA Starter Kit Board, on pourra continuer à développer des applications floues avec MIFS des structures homogènes et hétérogènes, comme une commande d'un processus par une redondance composée d'un régulateur classique PID et un moteur d'inférence floue pour analyser l'effet des causes communes du système hétérogène.

Un autre aspect de notre travail consiste en la possibilité d'embarquer des structures redondantes du contrôleur flou sur un seul FPGA pour répondre d'une part aux contraintes de sûreté de fonctionnement et d'autre part augmenter les performances de traitement en temps réel.

L'analyse a aussi montré que le niveau de sécurité d'inférence floue dépend non seulement du taux de défaillance du composant, mais aussi du facteur de couverture du diagnostic qui indique à quel point le MIFS est capable d'autosurveillance face à une éventuelle défaillance dangereuse, et de la nécessité d'une évaluation d'une éventuelle défaillance de cause commune qui peut invalider la redondance du MIFS.

On a aussi constaté que la fréquence du test d'inspection a un impact négatif si on le retarde d'une période de plus de 10 ans, et que la meilleure valeur de la probabilité moyenne de défaillance sur demande de la fonction de sécurité a été obtenue pour l'architecture 1oo2D.

Les résultats de calcul de la valeur de la probabilité moyenne de défaillance sur demande du moteur d'inférence floue sûr, par la méthode des blocs-diagrammes, par la méthode de l'arbre de cause et par le graphe de Markov montrent que chaque méthode a ses caractéristiques d'analyse et de quantification de la fonction de sécurité.

On perçoit que la méthode des blocs-diagrammes de fiabilité (BDF) modélise le système par des blocs de diagramme fonctionnel et permet une vue d'architecture système. Par contre la méthode de l'arbre des causes exige en plus de l'analyse fonctionnelle la détermination des défaillances dangereuses et les événements qui peuvent être associés qui causent la perte de la fonction de sécurité. Les résultats des trois méthodes sont presque similaires si on considère que le facteur β et le taux de couverture DC sont précis.

La méthode des graphes de Markov est la plus compliquée des trois et permet une analyse sur une longue période de façon à prédire le comportement futur du MIFS connaissant l'état présent du processus.

Notre flot de conception applique certains outils de Co-design. On a proposé une amélioration du modèle V pour des applications des systèmes embarqués à base du circuit FPGA.

On a utilisé aussi les exigences d'analyse de risque et de la quantification du régulateur flou selon la norme de sécurité 61508 [CEI 06], dont on a aussi proposé une modification des formules utilisées par l'analyse du bloc-diagramme de fiabilité pour le calcul de la probabilité moyenne d'une défaillance sur demande PFDavg.

On a constaté que l'évaluation de la performance du régulateur flou MIFS par bloc-diagramme de fiabilité (BDF) nécessite de prendre en considération l'imprécision des valeurs de couverture de diagnostic. On a aussi proposé de modéliser l'imprécision de ces paramètres par des intervalles. La valeur de couverture DC peut être représentée par un intervalle DC [Dcmin Dcmax].

De ce fait, les différents taux de défaillances dangereuses deviennent :

$$\lambda_{\max}^{DU} = \frac{\lambda}{2}(1-DC_{\min}) \quad , \quad \lambda_{\min}^{DD} = \lambda_{\min}^{SD} = \frac{\lambda}{2}DC_{\min} \quad ; \quad \lambda_{\min}^{D} = \frac{\lambda_{\min}}{2}$$

$$\lambda_{\min}^{DU} = \frac{\lambda}{2}(1-DC_{\max}) \quad , \quad \lambda_{\max}^{DD} = \lambda_{\max}^{SD} = \frac{\lambda}{2}DC_{\max} \quad ; \quad \lambda_{\max}^{D} = \frac{\lambda_{\max}}{2}$$

Ainsi on obtient pour la probabilité moyenne de défaillance sur demande PFDavg pour une structure un parmi un (1oo1) la formule suivante :

$$\begin{cases} PFD_{avg\min} = (\lambda_{\min}^{DU} + \quad \lambda_{\min}^{DD}) \times t_{\min\_CE} \\ PFD_{avg\max} = (\lambda_{\max}^{DU} + \quad \lambda_{\max}^{DD}) \times t_{\max\_CE} \end{cases}$$

De même, on obtient pour la probabilité moyenne de défaillance sur demande PFDavg pour une structure au moins un parmi deux (1oo2) la formule suivante :

$$\begin{cases} PFD_{avg\min} = 2((1-\beta_D)\lambda_{\min}^{DD} + (1-\beta)\lambda_{\min}^{DU})^2 t_{CE} t_{GE} + \beta_D \lambda_{\min}^{DD} MTTR + \beta\lambda_{\min}^{DU}(\frac{T_i}{2} + MTTR) \\ PFD_{avg\max} = 2((1-\beta_D)\lambda_{\max}^{DD} + (1-\beta)\lambda_{\max}^{DU})^2 t_{CE} t_{GE} + \beta_D \lambda_{\max}^{DD} MTTR + \beta\lambda_{\max}^{DU}(\frac{T_i}{2} + MTTR) \end{cases}$$

Cette méthode qu'on a nommée « évaluation de la performance d'un système dédié à une application par bloc-diagramme de fiabilité par des intervalles de précision ».

Les perspectives de ces travaux sont multiples. Tout d'abord, à court terme, nos projets de recherches consistent à effectuer des travaux d'optimisation au niveau d'extraction (*instruction level parallelism* ou ILP) répondant aux contraintes de sûreté fonctionnelle.

Il serait intéressant de développer une interface entre le logiciel Labview-FPGA et le logiciel Co-design Xilinx qui nous permette de récupérer le code VHDL pour une optimisation du placement et un ordonnancement des tâches.

Un autre travail prioritaire concerne l'amélioration et l'optimisation du modèle VHDL obtenu en tenant compte des spécificités des cibles FPGA utilisées.

À moyen terme, notre objectif est d'insérer un flot de conception des systèmes d'inférence floue répondant aux contraintes de sûreté de fonctionnement dans l'environnement graphique Labview-FPGA/Matlab-Simulink pour proposer un outil complet de Co-design. Nous envisageons aussi d'appliquer la méthodologie proposée avec d'autres modèles de contrôleurs flous et d'étendre ces études à d'autres architectures de type Quad (2oo4) par exemple.

# Bibliographie


[ADC 00] L.T. Limited, Datasheet of LTC 2604 Family, LT.

[AES 99] A.E. Summers. Common cause and common sense, designing failure out of your safety instrumented systems (SIS), ISA Transactions 38 (1999) 291±299.

[AJG 87] A. J. v. d. Goor, Testing Semiconductor Memories : Theory and Practice, Inc. New York, NY, USA. © 1991 : John Wiley & Sons, 1987.

[AMB 13] Pr Amami Benaissa, « Logique floue », chapitre « Fondement mathématique de la logique floue », université Abdelmalek Essaadi, Tanger, 2013.

[AMB 14] Pr Amami Benaissa, « Logique floue », chapitre « Logique floue : Système d'inférence floue », université Abdelmalek Essaadi,Tanger, 2013.

[ATH 99] A. Doumar, T. Ohmameuda, and H. Ito, "Design of an automatic testing for FPGAs", in Proc. IEEE Euro. Test Workshop, May 1999.

[BBM 07] B. Bouchon-Meunier, *la Logique floue*, ouvrage, ISBN : 978-2-13-056260-3, p. 50-55, 2007.

[BBO 81] B. Boehm Software Engineering Economics, Prentice-Hall, 1981.

[CEI 06] CEI 61508. Sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables relatifs à la sécurité. Commission électronique internationale, Genève, Suisse, 2000.

[DRR 00] Xilinx. Device Reliability Report, Firt Quarter May 13, 2013.

[DPY 11] Dayal, C., Pardeep, K., Yogesh, M., "FPGA implementation of a fuzzy logic based handoff controller for microcellular mobile networks", Revue International Journal of Applied Engineering Research, Dindigul vol. 2, vol. 1, 2011.

[DSF 11] D. Saptono, « Conception d'un outil de prototypage rapide sur le FPGA », thèse soutenue à l'université de Bourgogne, Bourgogne, 2011.

[EIB 06] The Field-Programmable Gate Array (FPGA): Expanding Its Boundaries, In Stat Market Research, avril 2006.

[FCI 11] F. Ciutat, *SIL Automatisme et sécurité, intégrité des fonctions automatisées de sécurité*, Fontvieille, Apta Édition, 2e édition, 2011.

[FEG 05] F.Cottet, E. Grolleau, *Systèmes temps réel de contrôle-commande*, Éditions Dunod, 2005.

[FUZ 00] Inform fuzzytech MCU. "General Purpose fuzzyTech Edition generates portable C code, edition 8".

[GBL 00] W. M.Goble, "Control Systems Safety Evaluation and reliability", Research Triangle Park, NC 27709, International Society of Automation, 3 Edition 2010.

[GBL 01] W. M. Goble and A.C brombacher, “Using a failure modes, effects and diagnostic analysis (FMEDA) to measure diagnostic coverage in programmable electronic systemes”, Reliability Engineering and System Safety, vol. 66, no. 2, p. 145-148, 1999.

[GBL 10] W. M.Goble, “Control Systems Safety Evaluation and reliability”, Research Triangle Park, NC 27709: International Society of Automation, 3 Edition 2010.

[GBL 11] W. M.Goble, “Control Systems Safety Evaluation and reliability”, Research Triangle Park, NC 27709, International Society of Automation, 3 Edition 2010, p. 312.

[GHY 08] Guo, H. and Yang, X. (2008). “Automatic creation of Markov models for reliability assessment of safety instrumented systems”. Reliability Engineering and System Safety, 93 : 807815.

[HDL 00] IEEE Standard VHDL Language reference Manual, IEEE-SA Standard Board Approved, New York, 30 January 2000.

[HDL 13] Andreas Mäder ; VHDL Kompakt, Fachbereich Informatik Universitär Hamburg; 2013.

[ISO 06] ISO 13849-1, Sécurité des machines — Parties des systèmes de commande relatives à la sécurité — Partie 1 : Principes généraux de conception, ISO, 2006.

[ISA 01] ISA TR84.0.0.2.Safety instrumented System, Safety integrity Level, Évaluations techniques. Part 1 Introduction, version 4, North Carolina, 1997.

[ISE 04] X. Xilinx, The ISE 10.1 Quick Strat Tutorial a hands-on learning tool, ISE 10.1 Quick Start, 2002-2008.

[ISM 09] XILINX, Isim User Guide, UG660 : XILINX, September 16, 2009.

[JBÖ 07] J. Börcsök, Elektronische Sicherheitssysteme Hardwarekonzepte, Modelle und Berechnung, Heidelberg: Hülthig GmbH & Co. KG, 2 Edition 2007.

[JQI 04] J. Qin, “A Brief Introduction to Application-Dependent FPGA”, Dept. of Electrical and Computer Engineering, 2004.

[KPY 98] Passino , K, Yurkovich, S., “Fuzzy Control”, Reading, MA: Addision-Wesley,1998.

[LAZ 65] L. A 154adeh, “Fuzzy Sets”, Information and Control, n°1338-353, 1965.

[LAZ 00] “Lotfi A. Zadeh”, le site de la faculté; College of Engineering, Electrical Engineering and Computer Science, University of California at Berkeley.

[LIM 03] Fernanda Lima, Luigi Carro, Ricardo Reis, “Designing Fault Tolerant Systems on SRAM-based FPGAs”, Proceedings of Design Automation Conference, 2003.

[MAM77] Mamdani, “Application of fuzzy logic to approximate reasoning using linguistic synthesis”, IEEE Transactions on Computers, n°1C-26 (12), p. 1182-1191, 1977.

[MER 08] M. Merabti, Hatime, « Étude des systèmes flous à intervalle », thèse de magistère, département électronique, université de Constantine, décembre 2008.

[MBJ 13] M. Basilio, « Robustesse par conception de cicuits implantés sur FPGA SRAM et validation par injection de fautes », thèse de doctorat, laboratiore TIMA, université de Grenoble, août 2006.

[MJM 04] M. B. Tahoori, E. J. McCluskey, M. Renovell, P. Faure, “A Multi-Configuration Strategy for an Application Dependent Testing of FPGAs”, Proc. VLSI Test Symp., 2004.

[MSB 08] Mechri, W., Simon, C., Ben Othman, K., « Approche par intervalles pour l’évaluation imprécise des systèmes Instrumentés de sécurité », Revue E-STA, vol. 8, n$^{o}$ 1, p. 44-52, 2011.

[MSB 10] Mechri, W., Simon, C., Ben Othman, K., and Benrejeb, M. (2010a). « Chaînes de Markov floues multiphases pour l’évaluation de la performance imprécise des systèmes instrumentés de sécurité », in la Sixième Conférence internationale francophone d’automatique, Nancy, CIFA 2010, Nancy.

[NOR 00] Norme 1364-2001, IEEE Standart verilog Hardware Description language, 2001.

[PJA 90] J. Ashenden, The VHDL Cookbook, First Edition, Australia: university of Adelaide, 1 Edition 1990.

[RDS 00] Rauzy, A.,Dutuit, Y., and Signoret, J.-P. “Assessment of safety integrity levels with fault trees”, in ESREL Estoril, Portugal.

[SKU 12] Salma, K, Prof Uma, K., “Design and implementation of Fuzzy logic controller for DC motor”, Revue International Journal of Emerging technology and Advanced Engineering, vol. 2, Issue. 8, 2012.

[SSN 86] SIEMENS AG, SN29500, “Reliability and Quality Specifications Failures Rates of Components", Siemens Technical Liaison and Standardisation, 1986.

[TAV 96] Tavernier, C. *Circuits logiques programmables*. Paris, Dunod, 1996.

[TKF 95] T. Liu, W. K. Huang, and F. Lombardi, “Testing of uncustomized segmented channel FPGAs”, in Proc. ACMInt. Symp. On FPGAs, Feb. 1995, p. 125-131.

[VIL 88] A. Villemeur, *Sûreté de fonctionnement des systèmes industriels*, Éditions Eyrolles, Paris, 1988.

[VER 00] P. Fischer, « Die Hardware Beschreibungssprache Verilog », Universität Heidelberg, 2006.

[WEI 98] Wei WU. « Synthèse d’un contrôleur flou par algorithme génétique : application au réglage dynamique des paramètres d’un système », thèse de doctorat de l’université de Lille 1, 1998.

[XIL 12] XILINX, “Data-Sheet Spartan-3E Family FPGA”, chez Product Specification, XILINX, 2012, p. 22.

[XIL 52] XILINX, “Spartan-3 FPGA Family Advanced”, XILINX, XAPP452 (v1.1), June 25, 2000.

[XSK 11] XILINX, Spartan-3E FPGA Starter Kit Board User Guide, UG230: XILINX, January 20, 2011.

[YFR 01] Y. Dutuit, F. innal, A. Rauzy, and J,-P. Signoret, “Probabilistic assessments in relationship with safety integrity levels by using fault trees”, Reliability Engineering and System Safety, vol. 93, n° 12, p. 1867-1876, 2008, 17th European Safety and reliability Conference.

[WIK 00] « Catastrophe de Seveso », Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc., date last updated: 14 October 2013.Web. Date accessed: 12 November 2013. <http://fr.wikipedia.org/wiki/Catastrophe_de_Seveso>

[MBA 11] Mohammed Bsiss, Benaissa. Amami “Safety Fuzzy Logic Controller of 1oo2 Architecture for FPGA Implementation”, **IJCSNS** International Journal of Computer Science and Network Security, Vol. 11, n° 4, April 2011.

[MBA 11] Mohammed Bsiss, Benaissa. Amami “Synthesizable Implementation of Safety Fuzzy Logic Controller of 1oo2 architecture in FPGA”, **IJCES** International Journal of Computer Engineering Science, Vol. 1, Issue n° 2, November 2011.

[MBH 12] Mohammed Bsiss, Ibrahim. Haj Baraka, Benaissa. Amami, “A quantified Safety Analysis for Safety Fuzzy Logic Controller 1oo2 Reliability Block Diagrams”, **IEEE** International Conference on Control Systems Computing and Engineering, 23-25 Nov. 2012, Penang, Malaysia.