

Module Master M1

Systèmes temps réel et Informatique Industrielle Chapitre IV : Bus de communication pour le temps réel

Projet n°01 d'exposé des étudiants de Master M1 Informatique

Présenté par : Prof. Kholadi Mohamed-Khireddine
Département d'Informatique
Facultés des Sciences Exactes
Université Echahid Hamma Lakhdar d'El Oued
Tél. 0770314924
Email. kholladi@univ-eloued.dz et kholladi@yahoo.fr
Site Web. www.univ-eloued.dz
<http://kholladi.doomby.com/> et <http://kholladi.e-monsite.com/>



IV – Bus de communication pour le temps réel

IV.1 - Introduction

Les bus de communications avec des systèmes extérieurs à la machine se font selon les types de réseaux de connexions comme sur la figure IV.1.

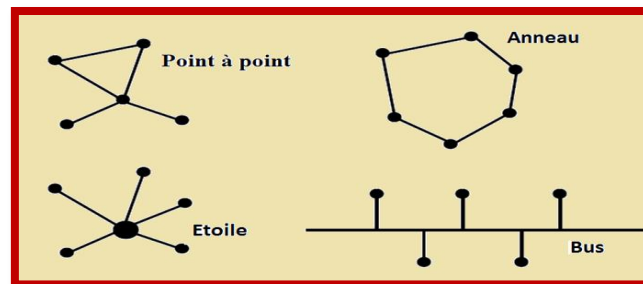


Figure IV.1 - Les types de réseaux de connexions

On a trois types de bus :

- Bus de communication : dans le bus de communications avec des systèmes extérieurs à la machine, on ne traitera pas les bus PCI, ISA, PCMCIA ou SCSI (périphériques), Ni les bus série comme RS232 ou USB, PC104 et PC104+ sont deux bus que l'on rencontre beaucoup dans les applications TR. Ils sont équivalents à ISA et PCI.
- Bus de terrain (CAN).
- Bus industriel (VME).

IV. 2 - Bus CAN

IV.2.1 – Introduction

Le Controller Area Network (CAN) présente en 1983 par la société Bosch, équipementier pour l'automobile comme la figure IV.2. Le site Web est <http://www.cancia.org>.

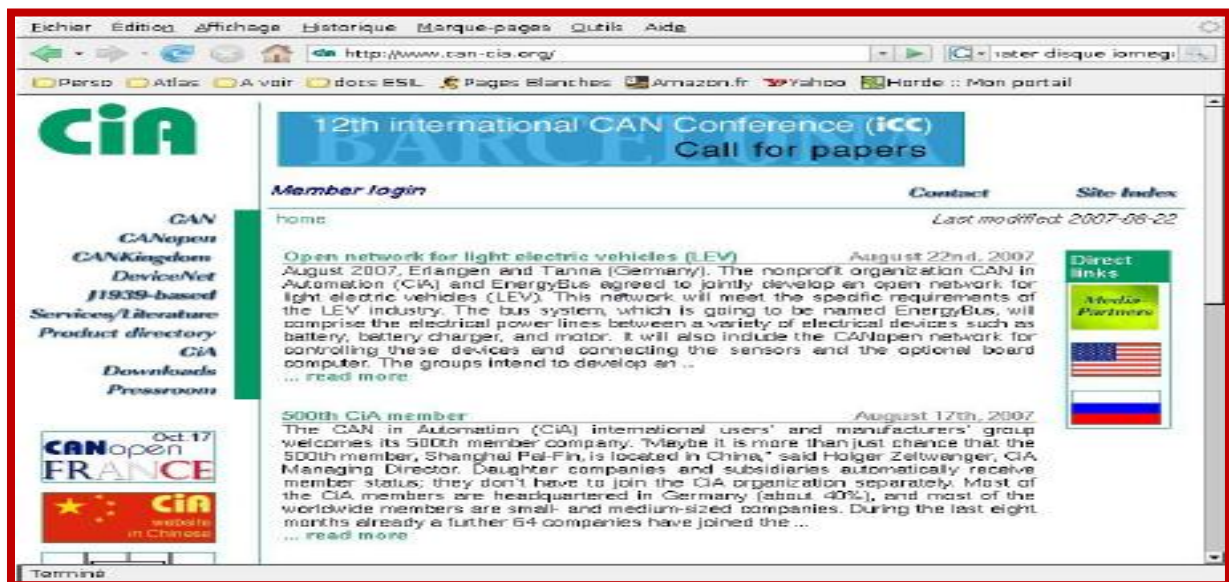


Figure IV.2 – Le site Web sur le bus CAN

Le protocole de communication défini par Bosch et Intel en 1985. La standardisation est faite par l'ISO en 1986. Le premier contrôleur (**82526 par Intel**) est apparu en 1987 et la première voiture multiplexée par CAN est apparue en 1991.

Le système basé sur un bus série multiplexe contient les caractéristiques suivants :

- Hiérarchisation des messages,
- Garantie des temps de latence,
- Souplesse de configuration,
- Réception de multiples sources avec synchronisation temporelle (CSMA/CA),
- Fonctionnement multi maître,
- Détection et signalisation des erreurs,
- Retransmission automatique des messages altérés dès que le bus est à nouveau au repos,
- Distinction des erreurs temporaires ou de non fonctionnalité permanente,
- Déconnexion automatique des nœuds défectueux.

IV.2.2 - Protocole de communication

La figure IV.3 illustre le protocole de communication par le mécanisme de communication par diffusion (broadcast) selon la norme ISO 11898.

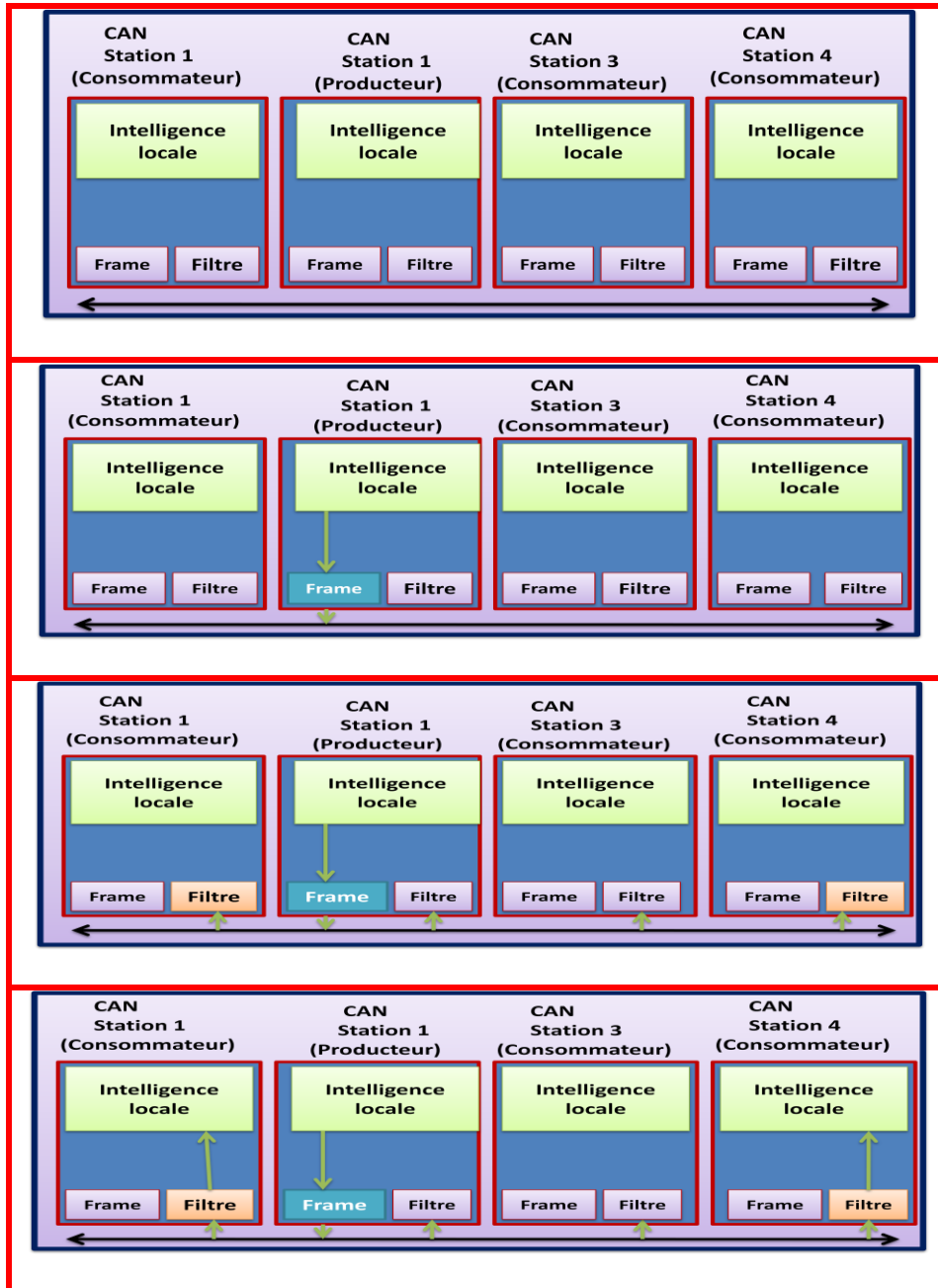


Figure IV.3 – Le mécanisme de communication par diffusion (broadcast)

La longueur physique du réseau impose la durée du bit (fréquence de transmission) avec une vitesse de propagation $\sim 5\text{ns/m}$.

a - Valeur du bus

Elle est indépendante du support physique (fil de cuivre, infrarouge, hertzienne, fibre optique, etc.). On a deux valeurs logiques : dominant (0) ou récessif (1). En cas de transmission simultanée d'un bit dominant et d'un bit récessif, la valeur résultante du bus sera dominant comme sur la figure IV.4.

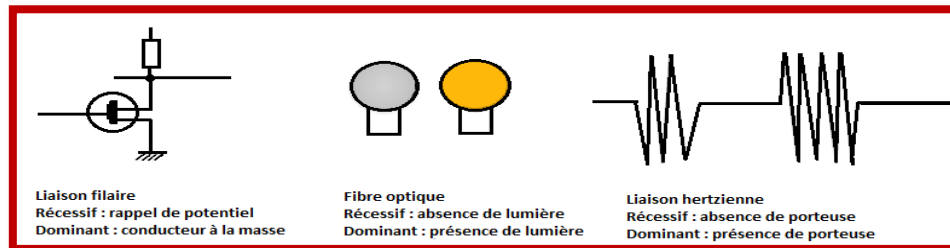


Figure IV.4 – Le support physique d'un bus

Procure une grande flexibilité (ajout de stations de réception).

Arbitrage à partir de l'identificateur du message :

- Comparaison bit à bit,
- L'identificateur de plus petite valeur est le plus prioritaire,
- Défini au moment du design et ne peut pas être changé dynamiquement.

Le tableau suivant donne le **format des messages standardisé**.

N°	Format des messages	Signification
1	DATA FRAME	Trame de données, transportant l'information d'un émetteur vers le récepteur.
2	REMOTE FRAME	Trame de requête, pour demander la transmission d'un DATA FRAME avec le même identificateur.
3	ERROR FRAME	Trame d'erreur transmise par une unité lorsqu'elle détecte une erreur de bus.
4	OVERLOAD FRAME	Trame de surcharge utilisée pour générer un retard supplémentaire entre DATA FRAME et REMOTE FRAME.

Le bus série est défini par sa durée (horloge). La figure IV5 illustre les trames de bourrages.

- Codage NRZ (No Return to Zero) : pendant la durée du bit, sa valeur reste constante, qu'il soit dominant ou récessif.
- Bit stuffing : après le passage de cinq bits à la même valeur, on émet un bit supplémentaire de valeur opposée.

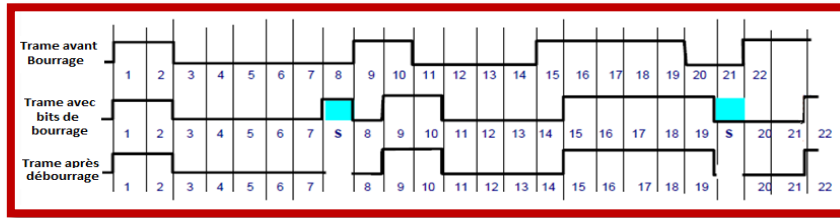


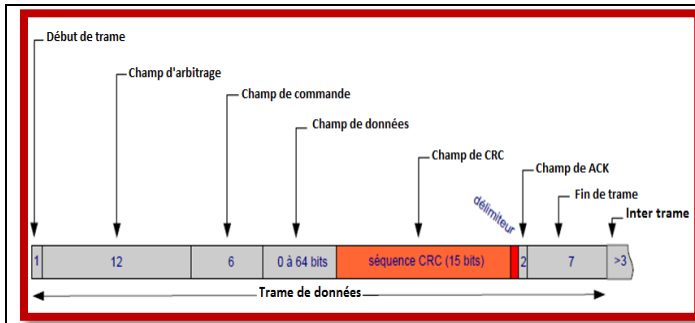
Figure IV.5 – Les trames de bourrage

b – Data Frame

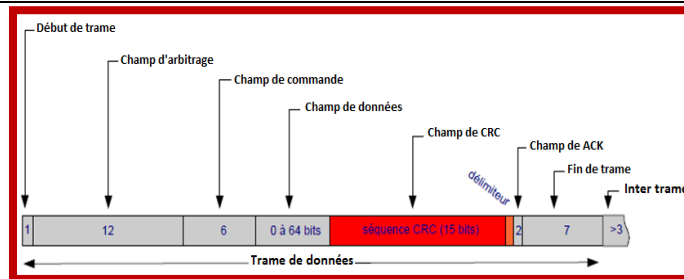
La figure IV.6 illustre l'explication des codes de chaque champ du Data Frame.

<p>The diagram shows a Data Frame structure with the following fields: IDLE, 1, 12, 6, 0 à 64 bits, 16, 2, 7, 3, IDLE. A double-headed arrow below the frame is labeled 'Trame de données'.</p>	<p>Data Frame départ</p>
<p>The diagram shows a Data Frame structure with an 11-bit arbitration field (ID10-ID0), 6, 0 à 64 bits, 16, 2, 7, and an inter-frame field (>3). A double-headed arrow below the frame is labeled 'Trame de données'.</p>	<p>Champ d'arbitrage :</p> <ul style="list-style-type: none"> • Onze bits pour l'identificateur : • Transmis de ID10 à ID0 (ID0 étant le bit le moins significatif), • Les sept bits les plus significatifs ne peuvent être tous récessifs.
<p>The diagram shows a Data Frame structure with an 11-bit arbitration field (ID10-ID0), 6, 0 à 64 bits, 16, 2, 7, and an inter-frame field (>3). A double-headed arrow below the frame is labeled 'Trame de données'.</p>	<p>Champ d'arbitrage :</p> <ul style="list-style-type: none"> • Onze bits pour l'identificateur, • Un bit RTR (Remote Transmission Request) : Dominant pour les trames de données et Récessif pour les trames de requêtes.

	<p>Champ de commande</p>
	<p>Champ de commande :</p> <ul style="list-style-type: none"> • Deux bits de réserve (comptabilisés avec la trame CAN2.08 et avec les futurs développements).
	<p>Champ de commande :</p> <ul style="list-style-type: none"> • Quatre bits donnant le nombre d'octets dans les champs de données
	<p>Champ de CRC</p>
	<p>Champ de CRC : séquence de quinze bits.</p>

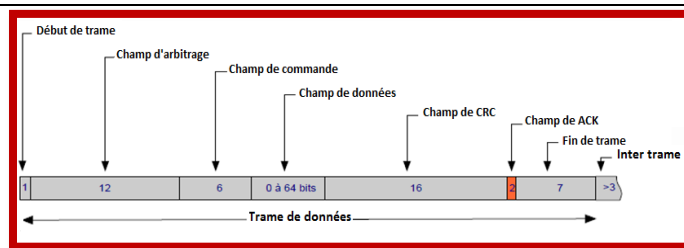


Champ de CRC : quinze bits + un bit délimiteur (récessif).



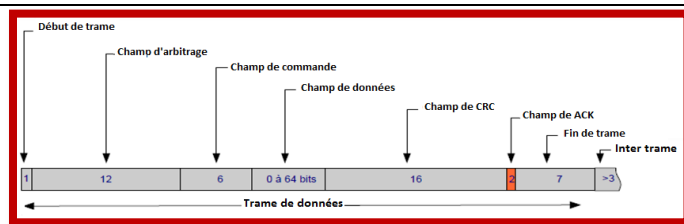
Champ de CRC : quinze bits + un bit délimiteur (récessif),

- Vérification de la validité du message transmis par un CRC suivant le code BCH,
- Cinq erreurs de bits indépendantes détectables (probabilité d'erreur résiduelle : $2^{-15} \sim 3 \times 10^{-5}$).



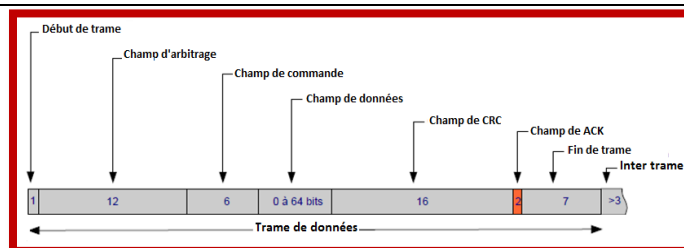
Champ de ACK : un bit ACK slot + un bit délimiteur,

- Deux bits récessifs émis par l'unité émettrice,
- Toute unité qui a reçu correctement le message (CRC) superpose un bit dominant sur le bit ACK slot, si Si OK, sur le bus : un bit dominant (ACK slot) entouré de deux bits récessifs (CRC delimitter et ACK delimitter).



Champ de ACK : un bit ACK slot + un bit délimiteur,

- L'envoi de l'ACK ne signifie pas que le récepteur est intéressé par le message.



Champ de ACK : un bit ACK slot + un bit délimiteur,

- Tout récepteur qui détecte une erreur et donc n'envoie pas d'acquiescement, doit ensuite envoyer une trame d'erreur.

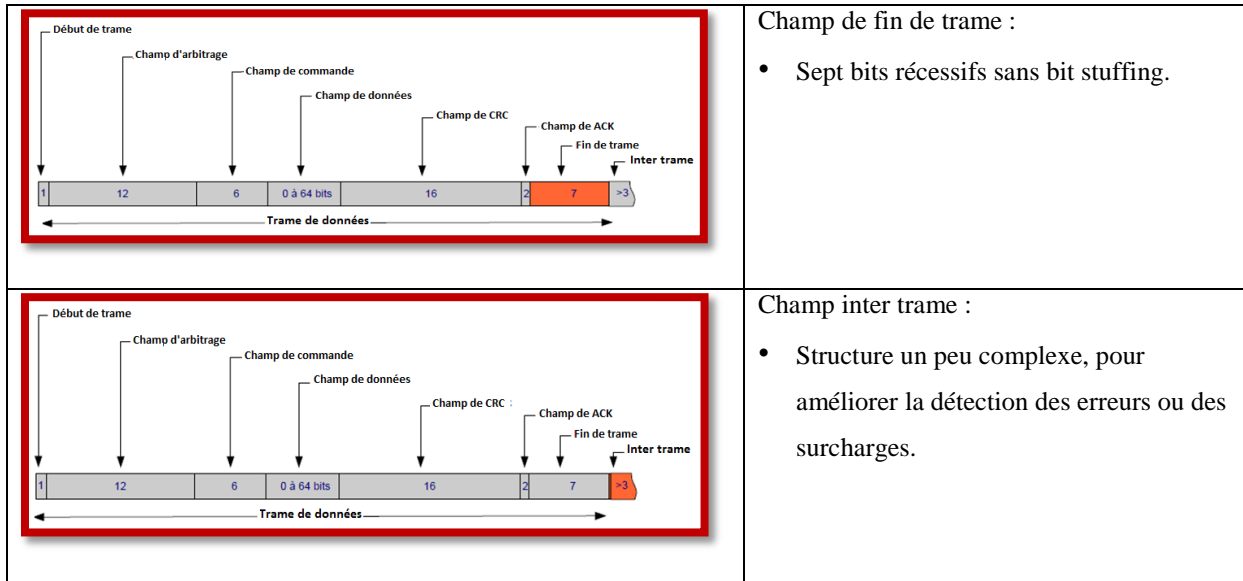


Figure IV.6 – Explication de la codification d'un Data frame

c - Remote frame

Pour demander la transmission de données par un autre nœud. On doit comprendre que la trame contient :

- Début de trame,
- Champ d'arbitrage,
- Champ de commande,
- Champ de **CRC**,
- Champ d'acquittement,
- Fin de trame.

À la différence de la trame de données, le bit RTR du champ d'arbitrage est récessif. À identificateur identique, la trame de donnée est prioritaire sur la trame de requête comme sur la figue IV.7.



Figure IV.7 – Nœuds de trame de donnée et nœud de trame requête

Les quatre derniers bits du champ de commande contiennent le nombre d'octets que devra contenir la trame de donnée satisfaisant la requête.

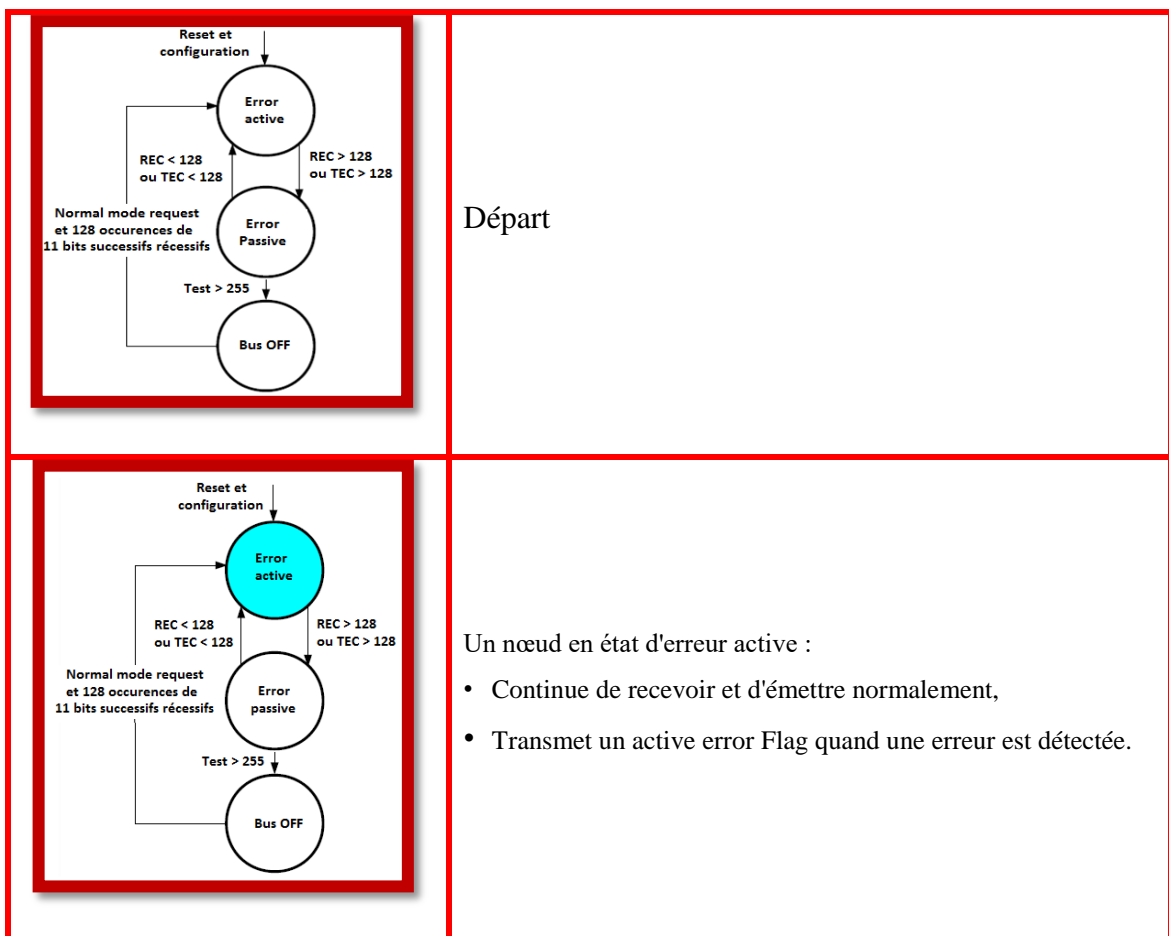
d - Détection des erreurs directement

Au niveau du message, on a l'utilisation d'un mécanisme de CRC (Cyclic Redundancy Check), la vérification de la structure globale de la trame et l'envoi d'un bit ACK par tous les receveurs qui fait qu'une erreur est détectée par l'émetteur si aucun ACK n'est reçu. Ou au niveau du bit, en monitorant au niveau des émetteurs les niveaux sur le bus et en comparant les bits émis et les bits reçus (permet en outre de vérifier la capacité de l'émetteur à détecter les erreurs). Les nombres d'erreurs détectées lors d'une transmission ou d'une réception sont stockés dans des compteurs (Transmit Error Counter et Receive Error Counter) :

- Si le message est transmis ou reçu correctement, le compteur décroît,
- Si une erreur est détectée, le compteur correspondant croît,
- Une pondération de huit est appliquée en défaveur de la décroissance des compteurs.

e - Analyse des erreurs détectées

La figure IV.8 illustre le diagramme du processus d'analyse des erreurs détectées.



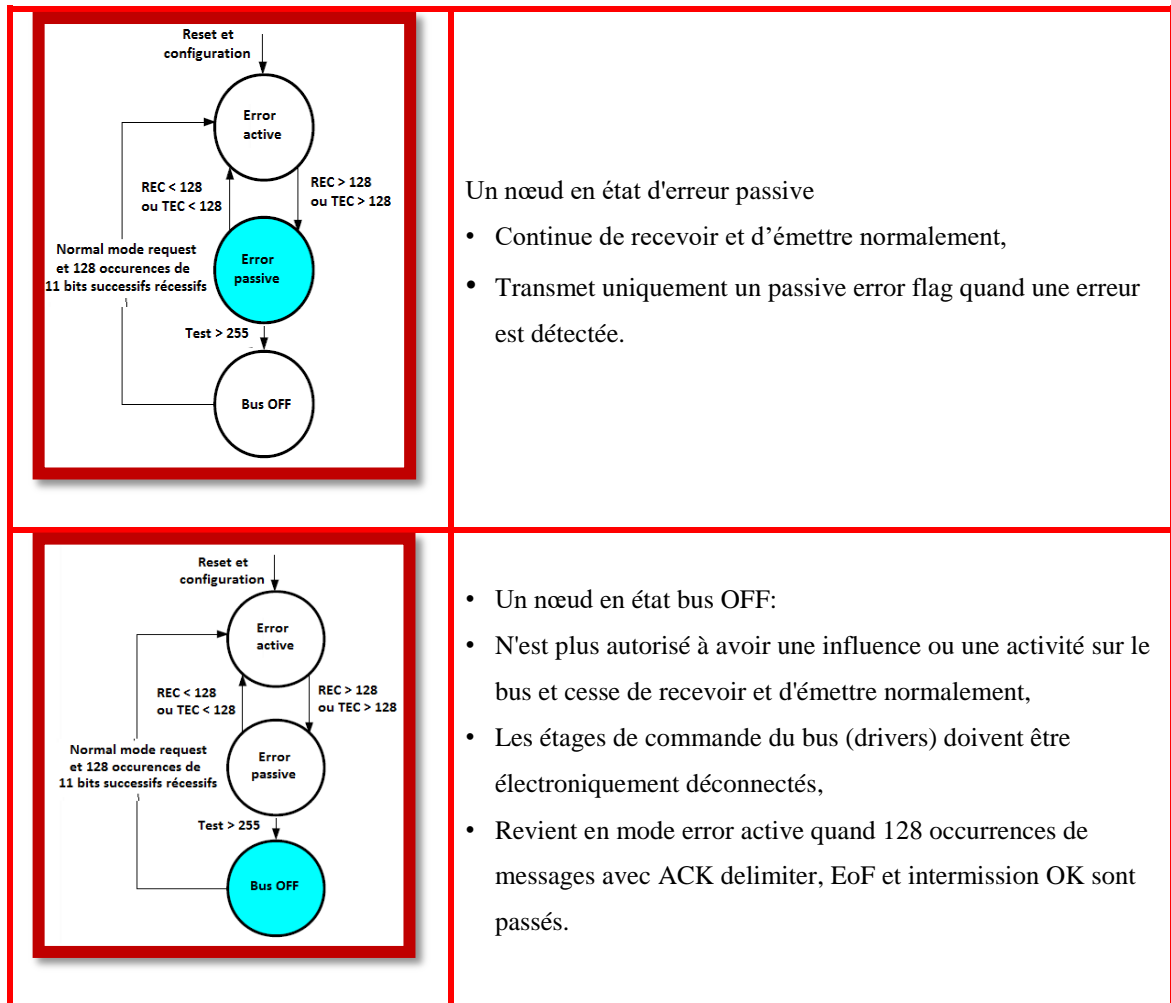


Figure IV.8 – Diagramme du processus d'analyse des erreurs détectées

Il existe un certain nombre de règles du jeu pour le comptage et le décomptage des erreurs selon que c'est un nœud émetteur ou récepteur qui détecte la faute. Ces règles font qu'un nœud fautif verra ses compteurs augmenter beaucoup plus vite que les autres nœuds et sera facilement repérable sur le réseau.

f - Détection des erreurs

Ici, on a la liste de détection des erreurs comme sur la figure IV.9 :

- Bit entaché d'erreur (comparaison du bit à émettre et du bit émis),
- Erreur de bit stuffing,
- Erreurs d'acquiescement,
- Erreurs de CRC,
- Erreurs de structure (observation d'un bit dominant là où on attendait un bit récessif) :
CRC delimiter, ACK delimiter, EoF, Error delimiter et Overload delimiter.

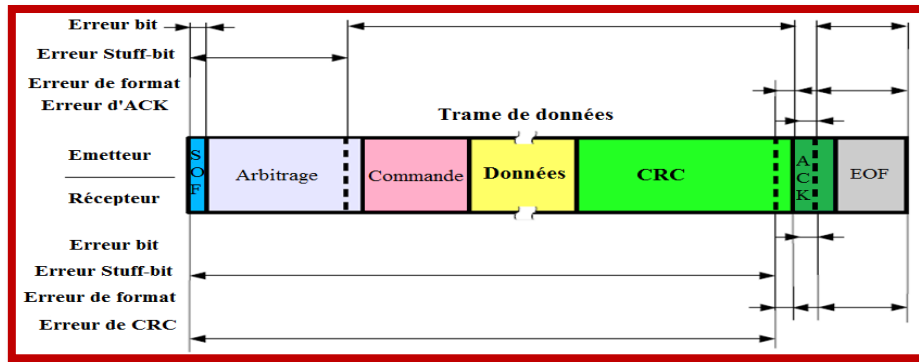


Figure IV.9 – Détection des erreurs

g - Signalisation des erreurs

La signalisation des erreurs pour informer les autres participants de la détection et pour envoyer un état des compteurs locaux (émission d'une active – ou passive – error flag). Le mécanisme d'émission d'une trame d'erreurs par la station détectrice : dès le bit suivant si l'erreur est due à un bit, un stuffing bit, une erreur de structure, une erreur d'acquiescement et après le ACK delimitier si c'est une erreur de CRC.

h - Trame d'erreurs

La figure IV.10 illustre la trame des erreurs. Dont le descriptif contient :

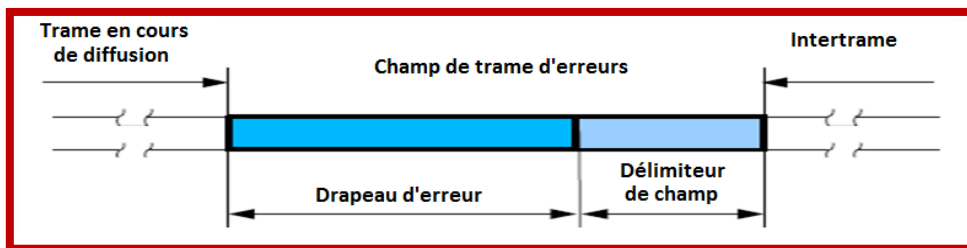


Figure IV.10 – Trame d'erreurs

- Champ des drapeaux d'erreurs,
- Superposition des drapeaux d'erreurs auxquels les différentes stations présentes sur le bus ont contribué,
- Active error flag : six bits dominants consécutifs,
- Passive error flag : six bits récessifs consécutifs (éventuellement écrasés par des bits dominants provenant d'autre nœuds),
- Délimiteur : huit bits récessifs consécutifs.

i - Récupération des erreurs

La récupération des erreurs d'effectue par réémission de la trame fautive, jusqu'à ce qu'elle passe correctement et qu'il n'y ait plus de message d'erreur. Les stations déficientes seront automatiquement éliminées lorsque leur compteur d'erreur les mettra en bus off, si toutes les stations sont en bus off → système bloqué → reset manuel !

IV.2.3 - Couche physique de communication

CAN suit le modèle OSI et implémente directement (au niveau du matériel) les deux couches inférieures : couche physique et couche de liens. Deux options pour la couche physique : haute vitesse (ISO 118982) et basse vitesse, tolérante aux fautes (ISO 118983).

a - Option à haute vitesse

La figure IV.11 illustre l'option à haute vitesse de la couche physique.

- De un Mb/s pour une longueur maximale de quarante m,
- À ~ cinquante kb/s pour une longueur maximale de un km,
- Sur un bus en paire torsadée terminée par des résistances de charge de 120 W.

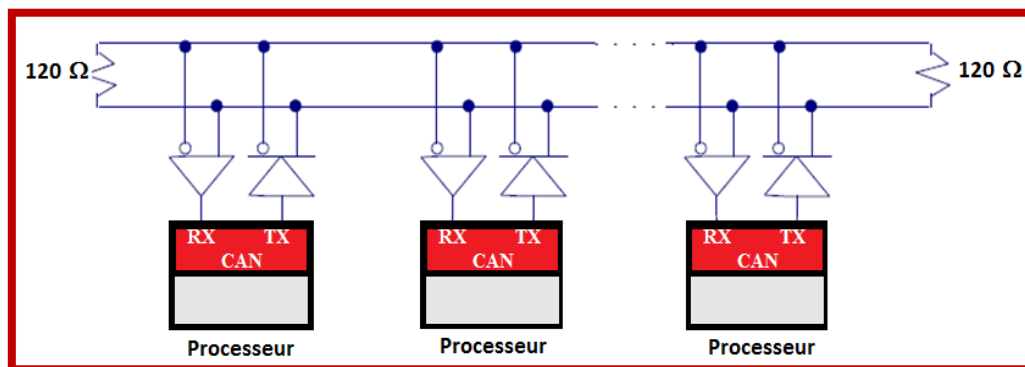


Figure IV.11 – Option à haute vitesse

b - Option à basse vitesse, tolérante aux fautes

- Plusieurs versions : ISO 115192, ISO 11992 et ISO 118983.
- Résistance de terminaison ~ cent W (pas moins).
- Entre vingt et trente deux stations sur un bus de quarante m maximum à cent vingt cinq kb/s maximum.

IV.2.4 - Extensions de CAN

TTCAN : time triggered CAN décrit dans la norme ISO 118984 pour gérer l'ordonnancement de messages dont l'envoi sur un bus CAN est déclenché par le temps ou par des événements, pour les applications de contrôle et pour l'amélioration des performances temps réel (diminution de la gigue dans la latence de transmission des messages, déterminisme du schéma de communication, tolérance aux fautes).

IV.2.4.1 - Extensions de CAN "TTCAN"

Il est basé sur l'utilisation d'un message de référence envoyé périodiquement par un maître du temps. L'envoi du message de référence démarre un cycle de base qui comporte plusieurs fenêtres temporelles de tailles différentes durant lesquelles des messages peuvent être envoyés :

- Fenêtres exclusives pour les messages périodiques, au début desquelles un message prédéfini est envoyé,
- Fenêtres arbitrées pour les messages non périodiques (spontanés),
- Fenêtres libres pour éventuellement étendre le réseau,
- Le pattern des fenêtres est défini au moment du design.

La tolérance aux fautes grâce à la présence de plusieurs maîtres du temps :

- Un seul maître effectif,
- Plusieurs maîtres potentiels,
- Accès à la maîtrise suivant la priorité du message de Référence,
- Tentative de prise de la maîtrise à l'occurrence d'un timeout sur la réception du message de référence.

La possibilité de changer dynamiquement la base de temps, ou même de passer temporairement à une structure aperiodique (event driven) grâce à un message de référence spécial.

IV.2.4.2 - Extensions de CAN : CANopen

C'est une couche de haut niveau construite au dessus de CAN. Initialement un projet ESPRIT, contrôle depuis 1995 par le consortium CiA (CAN in Automation). Il facilite le travail des concepteurs en proposant des objets pour la communication (Process Data Objects), des objets pour la configuration (Service Data Objects), des objets pour la gestion

du temps (réel), des objets pour la gestion du réseau et des couches applicatives permettant de réutiliser le software et le hardware.

IV.2.4.3 - Autres extensions de CAN

Pour les autres extensions de CAN, il faut voir les développements spécifiques, tels que : CAN Kingdom (contrôle de machines), <http://www.cankingdom.org/> et DeviceNet (automatisation d'usines) <http://www.odva.org/>

IV. 3 - Bus VME

IV.3.1 - Le bus VME

VME (Versa Module Eurocard) est un standard industriel né au début des années 1980, au début pour les cartes basées sur des processeurs Motorola 68000, très utilisé dans les applications mettant en jeu l'acquisition et le traitement des données. La norme passée dans le domaine public (pas de licence pour un bus propriétaire). Voir VME International Trade Union <http://www.vita.com>.

IV.3.2 - Le standard VME

Il est défini dans la norme : IEEE 1014-87 ou IEC 821-297. Il est conçu par des informaticiens industriels pour des informaticiens industriels. Les objectifs sont la communication entre deux unités sans gêner le fonctionnement interne des autres systèmes dont les performances dépendent essentiellement des unités et pas du bus lui-même et la grande latitude dans le choix des moyens d'optimisation des coûts et des performances sans nuire à la compatibilité. Le bus multiprocesseur asynchrone possède plusieurs cartes maîtresses peuvent accéder à des ressources communes (cartes mémoire, modules d'E/S, etc.). Il est cadencé par un mécanisme de handshake comme sur la figure IV12.



Figure IV.12 – le bus multiprocesseurs asynchrone

La cartes au format double Europe (norme DIN 41612 et DIN 41494) sur deux connecteurs P1 et P2 de quatre vingt seize broches comme sur la figure IV.13.



Figure IV.13 – La carte au format double Europe

a - Connecteur P1

C'est un bus de données de seize bits, un bus d'adresses de vingt quatre bits, six lignes de modification d'adresses et plusieurs lignes d'alimentation et de contrôle (voir <http://www.vita.com/vmefaq/vmepins.html>) comme sur la figure IV.14.

<i>Pin Assignment for the VMEbus P1/J1 Connector</i>				<i>Pin Assignment for the VMEbus P1/J1 Connector</i>			
Pin	Row a	Row b	Row c	Pin	Row a	Row b	Row c
1	D00	BBSY*	D08	17	GND	AM1	A21
2	D01	BCLR*	D09	18	AS*	AM2	A20
3	D02	ACFAIL*	D10	19	GND	AM3	A19
4	D03	BG0IN*	D11	20	IACK*	GND	A18
5	D04	BG0OUT*	D12	21	IACKIN*	GND	A17
6	D05	BG1IN*	D13	22	IACKOUT*	GND	A16
7	D06	BG1OUT*	D14	23	AM4	GND	A15
8	D07	BG2IN*	D15	24	A07	IRQ7*	A14
9	GND	BG2OUT*	GND	25	A06	IRQ6*	A13
10	SYSCLK	BG3IN*	SYSFAIL*	26	A05	IRQ5*	A12
11	GND	BG3OUT*	BERR*	27	A04	IRQ4*	A11
12	DS1*	BR0*	SYSRESET*	28	A03	IRQ3*	A10
13	DS0*	BR1*	LWORD*	29	A02	IRQ2*	A09
14	WRITE*	BR2*	AM5	30	A01	IRQ1*	A08
15	GND	BR3*	A23	31	-12 VDC	+5 VSTBY	+12 VDC
16	DTACK*	AM0	A22	32	+5 VDC	+5 VDC	+5 VDC

* indique que le signal est activé quand le niveau est bas

Figure IV.14 – L'assignement des pins au bus VME

b - Connecteur P2

Les extensions du bus de données (D15D31) et du bus d'adresses (A24A31). Elles sont définies par l'utilisateur : VSB, PCI et Autres, etc. Une extension du bus pour les processeurs soixante quatre bits a été standardisée. Elle utilise des connecteurs P1 et P2 à cent soixante broches et plus un connecteur optionnel P0 dédié aux E/S.

IV.3.3 - Organisation du bus VME

Le transfert de données en mode maître-esclave, de type DMA (sans intervention de la CPU) : asynchrone, non multiplexe, transferts sur des largeurs de mots de huit, seize ou trente deux bits, sept niveaux de priorité d'interruption, quatre niveaux d'arbitrage du bus de données et lignes spécialisées pour la détection rapide de défauts, le contrôle de bus et le contrôle de l'alimentation électrique. Le bus VME organisé en cinq sous bus : le Data Transfer Bus, le Data Transfer Arbitration Bus, le priority interrupt Bus, l'utility bus et le serial Bus. Il Décrit en termes de modules fonctionnels qui servent à gérer l'utilisation du bus de transfert de données (DTB) et conceptuels, parfois associés à du hardware comme sur la figure IV.15.

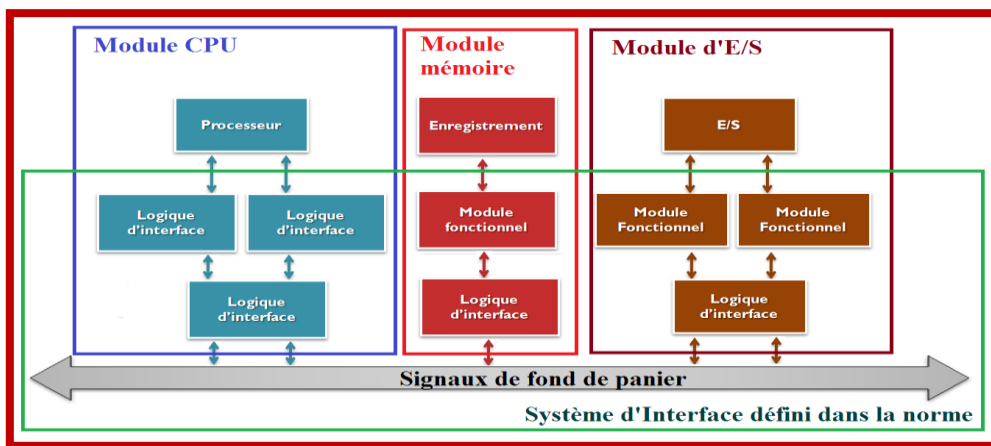


Figure IV.15 – Le système d'interface défini dans la norme VME

a - Fonctions d'interfaces :

Pour le transfert de données, les modules maîtres peuvent initier des transferts de données sur le DTB, les modules esclaves peuvent détecter les transferts et y participer et une carte peut posséder à la fois un module maître et un module esclave. Pour l'arbitrage du DTB, plusieurs maîtres peuvent coexister dans un système VME, un seul à la fois peut avoir le contrôle du bus, un module pour les fonctions de demande (requester) et un module pour les fonctions d'arbitrage (arbiter).

b - Gestion des interruptions

Une ressource peut demander une interruption de l'activité normale. Le module pour la génération de l'interruption (interrupter) et le module pour le traitement (interrupt handler).

c – Utilitaires

Il existe plusieurs utilitaire, tels que l'horloge, la (ré)initialisation, les défauts du système, le défaut d'alimentation et les Lignes série de communication.

IV.3.4 - Transfert des données

Par l'intermédiaire d'un bus spécialisé (DTB)

- Bus d'adresses :
 - Bus de trente deux bits A0 à A31,
 - Adressage sur seize, vingt quatre ou trente deux bits,
 - Espace mémoire adressable minimum : un octet.
- Bus de données :
 - Trois largeurs de mots : huit / seize / trente deux bits (D8/D16/D32).
- Bus de modification d'adresse :
 - AM0 à AM5.
- Lignes de contrôle de transfert par le maître.
- Lignes d'état contrôlées par l'esclave.

Lignes de modification d'adresse

Pour permettre au maître d'envoyer des informations supplémentaires lors d'un transfert :

1. Configuration dynamique du système, en imposant à l'esclave de ne réagir qu'à un seul code,
2. Sélection dynamique de l'emplacement de l'esclave dans l'espace d'adressage du maître,
3. Sélection du domaine d'adressage (court sur seize bits, standard sur vingt quatre bits, long sur trente deux bits),
4. Modification dynamique des privilèges nécessaires pour accéder à certains niveaux d'exécution de l'esclave,
5. Modification du type de transfert (par blocs) comme sur la figure IV.16.

A 5	A 4	A 3	A 2	A 1	A 0	M 5	M 4	M 3	M 2	M 1	M 0	fonctions	défini par
1	1	1	1	1	1	1	1	1	1	1	1	Accès ascendant en mode superviseur standard	Spécifié bus VME
1	1	1	1	1	1	0	0	0	0	0	0	Accès au programme en mode superviseurs standard	Spécifié bus VME
1	1	1	1	0	1	0	0	0	0	0	0	Accès aux données en mode superviseur standard	Spécifié bus VME
1	1	1	1	0	0	0	0	0	0	0	0	non défini	réservé
1	1	1	0	1	1	1	1	1	1	1	1	Accès ascendant en mode utilisateur standard	Spécifié bus VME
1	1	1	0	1	0	1	0	1	0	1	0	Accès au programme en mode utilisateur standard	Spécifié bus VME
1	1	1	0	0	1	0	0	0	0	1	0	Accès aux données en mode utilisateur standard	Spécifié bus VME
1	1	1	0	0	0	0	0	0	0	0	0	Non défini	réservé
1	1	0	X	X	X	X	X	X	X	X	X	Non défini	réservé
1	0	1	1	1	1	1	1	1	1	1	1	Non défini	réservé
1	0	1	1	1	1	0	0	0	0	0	0	Non défini	réservé
1	0	1	1	0	1	0	1	0	1	0	1	Accès entrée/sortie en mode superviseur court	Spécifié bus VME
1	0	1	1	0	0	0	0	0	0	0	0	Non défini	réservé
1	0	1	0	1	1	0	0	0	0	0	0	Non défini	réservé
1	0	1	0	1	0	0	0	0	0	1	0	Accès entrée/sortie en mode utilisateur court	Spécifié bus VME
1	0	1	0	0	0	0	0	0	0	0	0	Non défini	réservé
1	0	0	x	x	x	x	x	x	x	x	x	Non défini	réservé
0	1	x	x	x	x	x	x	x	x	x	x	Non défini	utilisateur
0	0	1	1	1	1	1	1	1	1	1	1	Accès ascendant en mode superviseur étendu	Spécifié bus VME
0	0	1	1	1	1	0	0	0	0	0	0	Accès au programme en mode superviseur étendu	Spécifié bus VME
0	0	1	1	0	1	0	0	0	0	0	0	Accès aux données en mode superviseur étendu	Spécifié bus VME
0	0	1	1	0	0	0	0	0	0	0	0	Non défini	réservé
0	0	1	0	1	0	0	0	0	0	0	0	Accès ascendant en mode utilisateur étendu	Spécifié bus VME
0	0	1	0	1	0	0	0	0	0	0	0	Accès au programme en mode utilisateur étendu	Spécifié bus VME
0	0	1	0	0	1	0	0	0	0	0	0	Accès aux données en mode utilisateur étendu	Spécifié bus VME
0	0	1	0	0	0	0	0	0	0	0	0	Non défini	réservé
0	0	0	x	x	x	x	x	x	x	x	x	Non défini	réservé

Figure IV.16 – Les modificateurs

Lignes de contrôle et d'état

- Quatre lignes de contrôle pour le module maître :
 - Address strobe $\rightarrow e \overline{AB}$
 - Data strobe zéro : sélection l'octet de poids faible $\rightarrow \overline{DS0}$
 - Data strobe un : sélection l'octet de poids fort $\rightarrow \overline{DS1}$
 - Accès à un mot de trente deux bits $\rightarrow \overline{LWORD}$
 - Direction du transfert (niveau bas : écriture maître \rightarrow esclave) $\rightarrow \overline{WRITE}$
- Deux lignes d'état contrôlées par le module esclave :
 - Data acknowledge (transfert réussi) $\rightarrow \overline{DTACK}$,
 - Erreur dans le transfert en cours $\rightarrow \overline{BERR}$.

Bus d'adresses

Pour accéder à un octet de mémoire par le module maître :

- Valeur de l'adresse sur le bus d'adresse **A02A32**,

- Utilisation des lignes DS0, DS1, LWORD et A01 pour le mode d'accès.

Lignes de données

- Sur les lignes D0 à D15,
- Les lignes DS0, DS1, A01 et LWORD permettent de choisir l'octet, la longueur du transfert et l'alignement.

Exemple de lecture comme sur la figure IV.17.

1. le module maitre utilise les lignes A01-31, AM0-AM5 et LWORD pour choisir l'esclave et le mode.
2. le maitre désactive IACK pour indiquer que ce n'est pas un cycle en réponse à un interrupt.
3. le maitre active AS pour valider l'adresse.
4. le maitre désactive WRITE (niveau haut) pour indiquer un cycle de lecture.

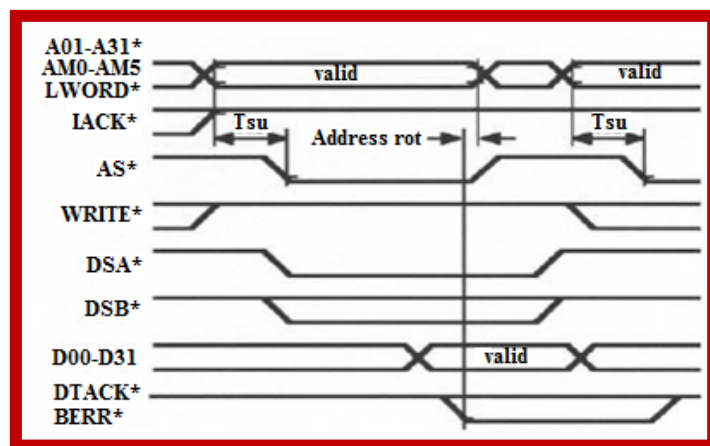


Figure IV.17 – Exemple de lecture

IV.3.5 - Arbitrage pour l'accès au bus

Plusieurs modules maitres peuvent coexister dans le même châssis. Un seul maitre peut posséder le contrôle du DTB. La logique d'accès est implantée au niveau du hardware :

- Accès exclusif d'un maître au DTB,
- Gestion des demandes d'accès (BUS REQUEST) :
 - Trois modes :
 - PRIARBITER : priorité décroissante de BR3 à BR0,
 - RRSARBITER : Round robin de BR(n) à BR(n1),
 - SGLARBITER : requêtes de BR3 seulement acceptées.

Lignes d'arbitrage du bus DTB

Six lignes de bus et quatre lignes en daisychain :

- Quatre lignes de requêtes BR0 à BR3,
- Une ligne BBSY (Bus Busy) pour indiquer que le bus est contrôlé par un maître,
- Une ligne BCLR (Bus Clear) pour indiquer qu'une requête de priorité supérieure est présente en mode PRIARBITER,
- Des lignes d'allocation de requête BG[03] IN et BG[03] OUT qui doivent être transmises d'un module à l'autre par une daisy chain (ou par des jumpers s'il n'y a pas de module présent ou si le module n'utilise pas ce niveau).

Un module demandant le contrôle du bus doit donc gérer une ligne de demande (BR[03]), Une ligne d'allocation (BG[03] OUT) et la ligne de bus busy BBSY. Le module assurant le rôle d'arbitre DOIT occuper le slot 1 dans le châssis VME. Le positionnement géographique des modules procure implicitement un niveau de priorité supplémentaire comme sur la figure IV.18.

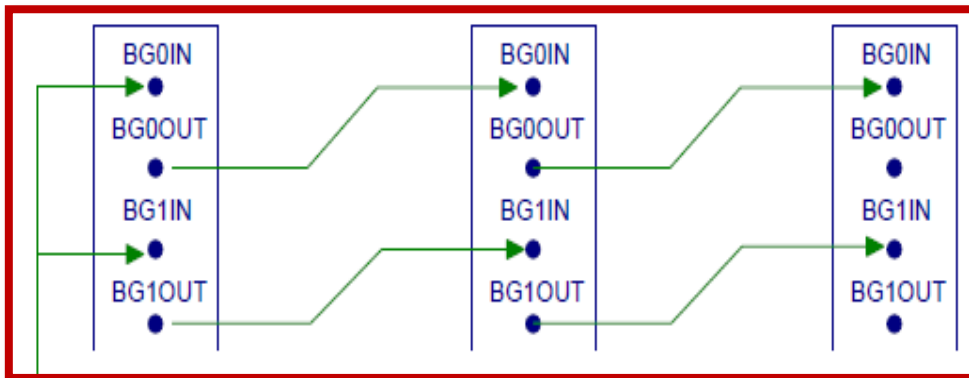


Figure IV.18 – Module de contrôle du bus pour la gestion des lignes de demande

Générateur de requête

Il gère le signal DEVICE_WANT_BUS du maître de son module VME, ou l'unité de gestion des interrupts de ce même module. Il génère un Bus Request (BR[03]). Il détecte le signal BG[03] IN et le transmet à BG[03] OUT si le bus n'est pas demandé par le maître de son module. Si l'unité maître demande le DTB et que la ligne BG[03] IN est en niveau bas, le générateur indique la disponibilité du bus par un signal DEVICE_GRANTED_BUS et met la ligne BBSY en niveau bas.

Le tableau suivant illustre les trois modes de fonctionnement pour l'unité de requête de bus :

RWD	Release When Done	BSSY est remise au niveau haut dès que la demande de contrôle est annulée.
ROR	Release On Request	BSSY n'est remise au niveau haut que quand la demande de contrôle est annulée et qu'une autre requête BR[03] est présente (évite la surcharge du bus).
FAIR	Pas de requête de bus tant que des requêtes sont encore en cours de traitement.	

IV.3.6 - Les modes de transfert de données

Le cycle VME assure une séquence d'opérations qui permet la communication entre deux modules sur le bus VME, en suivant les protocoles définis dans la norme. Une fois qu'un cycle est démarré, et jusqu'à ce qu'il se termine, les autres cartes du châssis n'ont plus accès au contrôle. Les transferts de données peuvent se faire en modes huit bits (octet et octet étendu), seize bits et trente deux bits. Le tableau suivant nous donne les sept types de cycles.

N°	Types de cycles	Signification
1	Lecture	Transfert esclave → maître.
2	Écriture	Transfert maître → esclave.
3	Lecture par blocs	Seule l'adresse de départ est fournie et le maître ne rend le bus qu'à la fin deux cent cinquante six blocs maximum.
4	Écriture par blocs	Analogue à la lecture par blocs.
5	Lecture-modification-écriture	Sans rendre le bus pour émuler l'atomicité de l'accès à une variable.
6	Adressage	Pour positionner simplement les adresses sur le bus.
7	Acquittement d'interruption	Pour confirmer la réception d'un interrupt et le transfert du vecteur.

IV.3.7 - Bus hiérarchisé d'interruptions

Le bus hiérarchisé d'interruption est utilisé pour générer et traiter les interruptions. Deux catégories de systèmes de gestion comme sur la figure IV.19. Le système centralisé où un seul processeur reçoit et traite les interruptions.

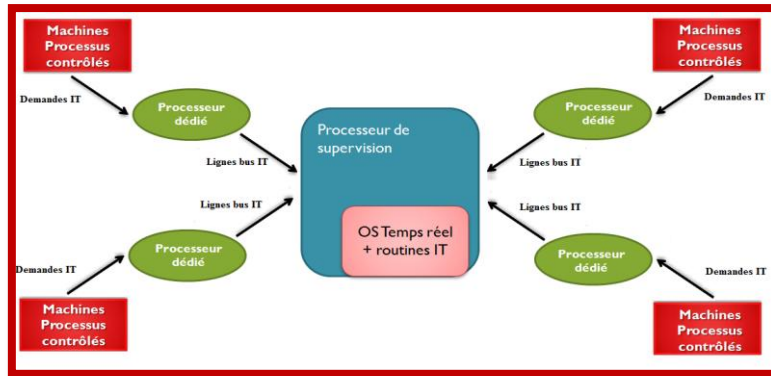


Figure IV.19 – Bus hiérarchisé d'interruptions

Le système décentralisé où chaque processeur ne traite que les interruptions qui lui sont destinées. L'allocation de la prochaine tâche se fera sur le premier processeur qui se libère. Chaque processeur exécute une partie du système d'exploitation et a accès aux ressources globales comme sur la figure IV.20.

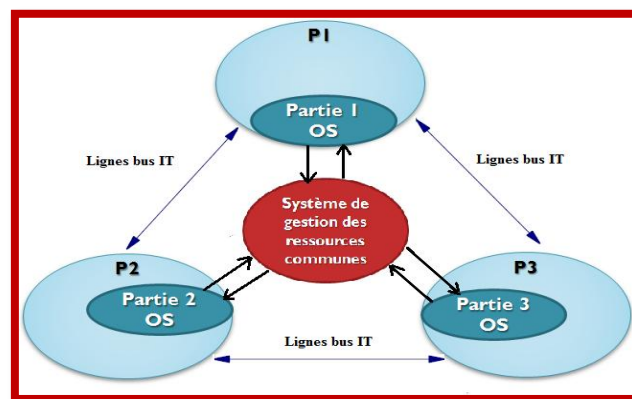


Figure IV.20 – Le système décentralisé

Lignes de gestion des interruptions

- Sept lignes de demande IRQ1 à IRQ7 :
 - Chaque ligne peut être activée par un niveau bas de l'unité de demande d'interruption (interrupter)
 - IRQ7 est la plus prioritaire dans un système centralisé
- Une ligne de prise en compte IACK.

- Deux lignes IACKIN et IACKOUT pour faire un chaînage des interruptions :
 - La continuité des lignes doit être assurée par des cavaliers si un slot n'est pas occupé.

Unité de gestion des interruptions, pour :

- Déterminer parmi les demandes celle qui a la priorité la plus élevée.
- Demander le contrôle du bus via l'unité de demande de contrôle du bus et générer alors le signal IACK de prise en compte de l'interrupt sur le bus.
- Lire le vecteur d'interruption et initialiser la séquence de traitement.

Unité de demande des interruptions

- Adresse une demande à l'unité de gestion associée à la ligne de demande utilisée.
- Fournit un vecteur d'interruption quand elle reçoit le signal de prise en compte.
- Transmet le signal de prise en compte si elle n'est pas la source.

IV.3.8 - Bus utilitaires

Le tableau suivant traite les lignes utilitaires.

SYSCLK	Horloge système.
SERCLK	Horloge série.
SERDAT	Données série.
ACFAIL*	Panne d'alimentation.
SYSRESET	Initialisation du système (prise en compte optionnelle).
SYSFAIL	Panne système.

Modules utilitaires

- Générateur de l'horloge système :
 - Autonome, fréquence seize MHz, dans le module dans le slot #1,
 - Base de temps pour certains systèmes.
- Horloge série :
 - Forme suivant IEEE P1132,
 - Pour des transferts série, en utilisant également les deux lignes de transfert série.

Surveillance de l'alimentation électrique

La figure IV.21 illustre la surveillance de l'alimentation électrique.

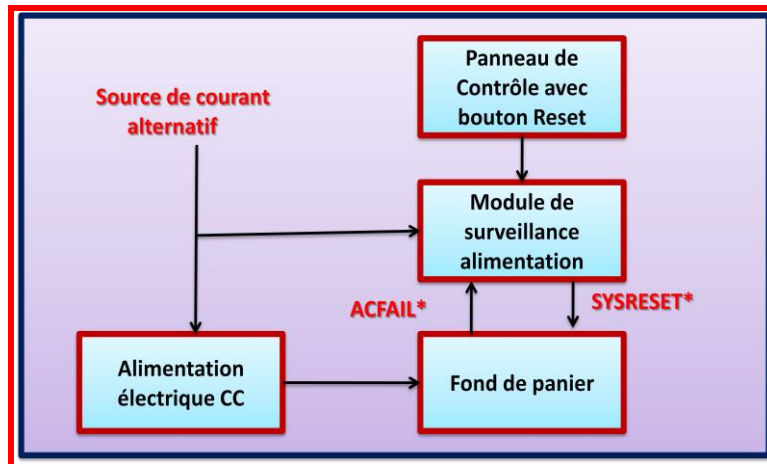


Figure IV.21 – Surveillance de l'alimentation électrique

IV.3.9 - Les principales cartes VME

Cartes processeurs

- Assurent généralement aussi les fonctions de maître.

Cartes mémoires

- De moins en moins utilisées avec l'augmentation de la mémoire disponible sur les cartes processeurs.

Modules d'entrées-sorties

- Liaison avec le monde extérieur,
- Prise en compte des interruptions extérieures,
- Liaison avec d'autres types de bus.