MASTER EN SCIENCES INFORMATIQUES

Conception et réalisation d'un contrôleur de bus I.E.E.E.-488

Vanobberghen, William

Award date:
1984

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur.
Institut d' Informatique.

Année Académique 1983-1984.

Conception et Réalisation d' un contrôleur de bus I.E.E.E.-488.

Vanobberghen William.

Mémoire présenté en vue de l' obtention du grade de Licencié et Maître en Informatique.

Promoteur de mémoire : Mr. J. Brunin.

A Monsieur J. Brunin, promoteur de mémoire, qui a bien voulu accepter de diriger cette étude, en témoignage de respect pour son expérience et ses précieux conseils,

Monsieur J. Demarteau, qui nous a permis de réaliser et de mener à bien ce projet, en remerciement de sa patience,

tous les autres, pour leur soutien,

nous adressons nos sentiments de profonde reconnaissance.

CHAPITRE I. LA NORME I.E.E.E.-488.

c = ç si pas Cécile.

CHAPITRE II. Conception et réalisation d' un contrOleur de bus I.E.E.E.-488.

AVANT-PROPOS

Le mémoire que nous présentons ici, traite des problèmes d' interfacage relatifs au domaine de l' instrumentation.

Cette étude s' appuie sur l' évolution récente de l' intégration d' appareils de mesure au sein de configurations de traitement de données, et fait part des efforts entrepris en vue d' aboutir à la reconnaissance de règles et de méthodes d' interfacage universellement acceptées, et à leur sanction par une norme.

Ce travail bénéficie de l' expérience acquise au cours des premières années de travail passées au sein d' une équipe familiarisée aux problèmes d' adaptation et de connexions entre systèmes de traitement de données.

Alors que l' étude d' un premier sujet de mémoire traitant de protocoles de transmission et de primitives d' échanges de données avait été entrepris, il fut décidé, en accord avec le promoteur de mémoire, d' orienter cette étude vers un domaine plus spécifique, celui de l' instrumentation, pour lequel un certain intérêt nous avait été signifié, lors de contacts menés auprès d' un distributeur d' instruments de mesure.

Il fut alors envisagé d' étudier un produit particulier, le contrôleur de bus I.E.E.E.-488, suceptible de répondre aux souhaits d' utilisateurs confrontés au problème de l' agencement d' appareils de mesure au sein de petites configurations, et, ce, à moindre cout.

Ce projet nous apparut intéressant à d' autres égards.

La révision de la norme I.E.E.E.-488 en 1978, et la présentation de directives relatives aux contenus et formats de données, publiées en 1982, constituèrent, selon nous, un sujet d' actualité. La publication de ces recommandations, trop récente pour pouvoir faire l' objet d' une étude complète dans ce travail, a cependant été abordée, et donne lieu à une synthèse que nous reprenons en annexe du document.

L' évolution de l' électronique digitale, d' une manière générale, et des composants électroniques associés aux micro-processeurs, en particulier, nous permis d' envisager le recours à des circuits spécialisés dans la gestion du bus, offrant l' avantage de la souplesse de développement, tant sur le plan matériel que logiciel, ainsi que de meilleures performances globales, tout en constituant une solution économiquement avantageuse.

CHAPITRE I. LA NORME I.E.E.E.-488.

1.0. Introduction.

La conception et la réalisation d' un contrôleur de bus I.E.E.E.-488, s' appuie sur une étude approfondie des caractéristiques du bus et du texte de la norme.

Cette étude est entreprise dans la première partie de ce travail.

Après un apercu des systèmes d' interfacage et de leur évolution dans le temps, nous envisageons l' étude du bus et de son fonctionnement logique, ainsi que l' analyse des messages qui y sont associés.

L' interface au bus est ensuite analysée sous quatre aspects distincts.

L' aspect fonctionnel définit l' ensemble des fonctions d' interfacage, ainsi que leur logique de fonctionnement, indépendamment des caractéristiques fonctionnelles de l' appareil. Cette présentation utilise les formes de représentation des fonctions d' interfacage issues de la norme, ainsi qu' une présentation originale, illustrant systématiquement chacune des fonctions sous forme d' un automate séquentiel.

L' aspect électrique décrit les caractéristiques électriques de l' interface et, notamment, les niveaux en courant et en tension à respecter.

L' aspect mécanique concerne l' assemblage des appareils au sein d' une configuration, et définit les types de connecteur utilisés, ainsi que le brochage du cable de liaison.

L' aspect l' opérationnel définit les caractéristiques de l' interface, en ce qui concerne le fonctionnement de l' appareil proprement dit. En raison du caractère trop spécifique de ces caractéristiques, et conformément au principe d' indépendance, voulant que les parties fonctionnelle et d' interfacage d' un appareil soit dissociées, la norme a cru bon, de ne pas devoir définir cet aspect de l' interface.

Le dernier chapitre est alors consacré au contrôleur proprement dit. L' étude du contrôleur est entreprise, en étudiant de manière exhaustive, chacun des états associé à la fonction de contrôle. Des représentations de configurations d' appareils illustrent les échanges de données susceptibles de se produire entre appareils, en présence ou non d' un contrôleur de bus.

1.1. Apercu des systèmes d' interfacage relatifs au domaine de l' instrumentation.

Le souci de concevoir une interface digitale universelle relative à l'utilisation d'instruments de mesure programmables, est né de l' évolution constante des instruments de mesure depuis les années 60, et de la difficulté de développer des systèmes permettant une communication éffective entre les instruments composants ces systèmes.

1.1.0. Généralités.

Les premiers instruments de mesure programmables ne permettaient l'introduction de commandes qu' au moyen de boutons poussoirs. Ces appareils ne possédaient pas la faculté de mémoriser les commandes ainsi introduites, et produisaient les résultats de mesure sous forme codée BCD ou binaire. L' absence de conventions dans l'interprétation des signaux électriques (adoption conjointe de la logique positive et de la logique négative) accentuait la difficulté de mettre en relation deux ou plusieurs appareils échangeant une même information de type digital.

A coté de ces incompatibilités, subsistaient des problèmes relatifs à une présentation des résultats propre$ à chaque instrument, la présence de lignes d' entrée et de sortie distinctes et de type unidirectionnelles, et une programmation de chaque instrument, eu égard à ses caractéristiques propres.

Un pas dans le sens de l' évolution apparut dans le courant des années 50, avec l' apparition des premiers bus de transfert d' information digitale.

La possibilité était alors donnée, de pouvoir connecter un compteur de fréquence à un enregistreur digital, au moyen d'un ensemble de lignes électriques parallèles et unidirectionelles, sans aucune capacité de programmation des appareils mis en présence.

Le succès de ces premières tentatives s' avèra important, et donna lieu au développement d'un nombre important d' appareils offrant une interface digitale.

Avec un intéret croissant marqué pour les premiers appareils programmables, il devint opportun de concevoir des appareils susceptibles de supporter des flux d'information de type bidirectionnel.

A cette nouvelle contrainte allait s' ajouter celle de pouvoir intégrer plusieurs instruments au sein d'une même configuration-système.

Le contrôleur apporta une solution à ce problème.

Au départ, le contrôleur s' insérait à l'intérieur d'une structrure en étoile.

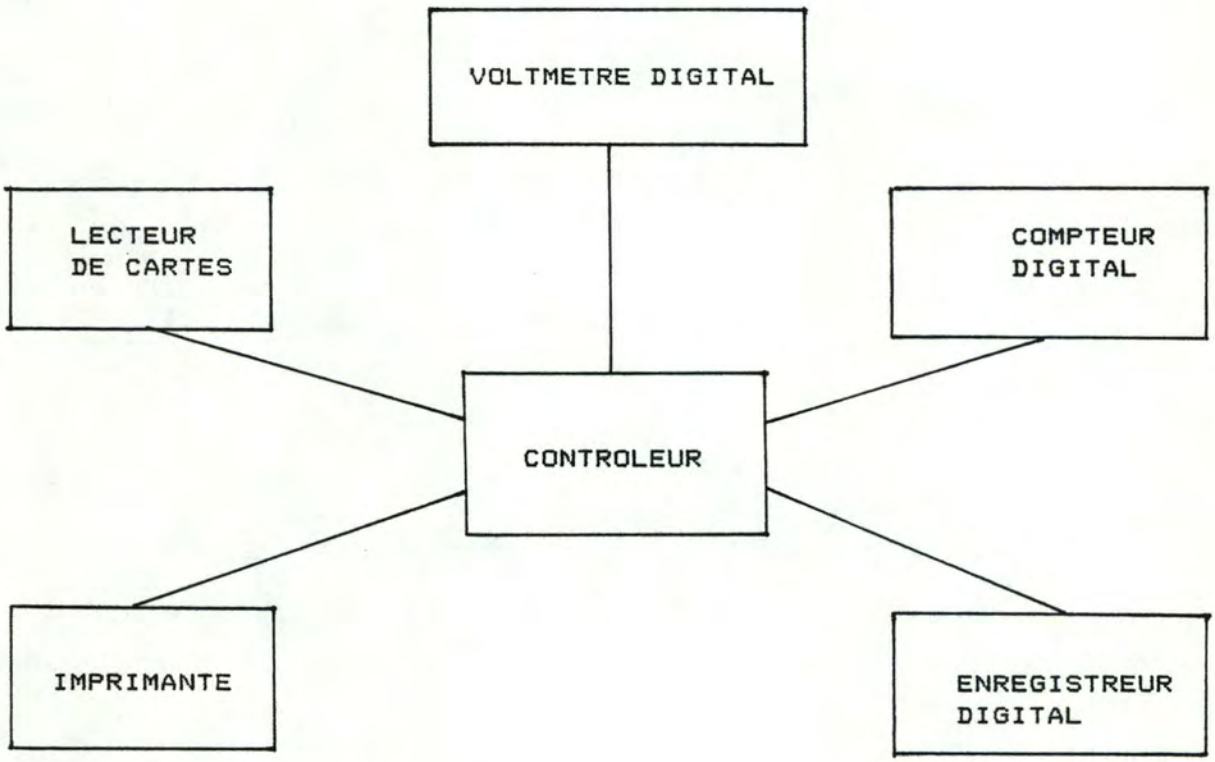


Fig. 1.1.0.0. Contrôleur opérant à l' intérieur d'une configuration en étoile.

Ce type de configuration intégrait le plus souvent un amalgame d' appareils de mesure standard, ainsi que des appareils développés pour les besoins de l' application rencontrée.

Chaque appareil proposait ses propres interfaces et conventions de transfert, sans souci de concordance avec d' autres instruments.

Une étape ultérieure dans cette évolution fut franchie avec l' apparition de petites configurations de systèmes de type autonome et comprenant un appareil intégrant les fonctions de contrôle.

Ces nouveaux systèmes tentèrent de normaliser les lignes électriques de transfert, notamment en définissant les niveaux de courant et tension de ces lignes, ainsi que la signification logique à attacher aux niveaux ainsi définis.

Les transferts de données se réalisèrent à partir d' une codification commune aux appareils de la configuration, et selon un même format. Le caractère unidirectionnel de ces lignes ne permit toutefois le transfert que d'un seul type d' information.

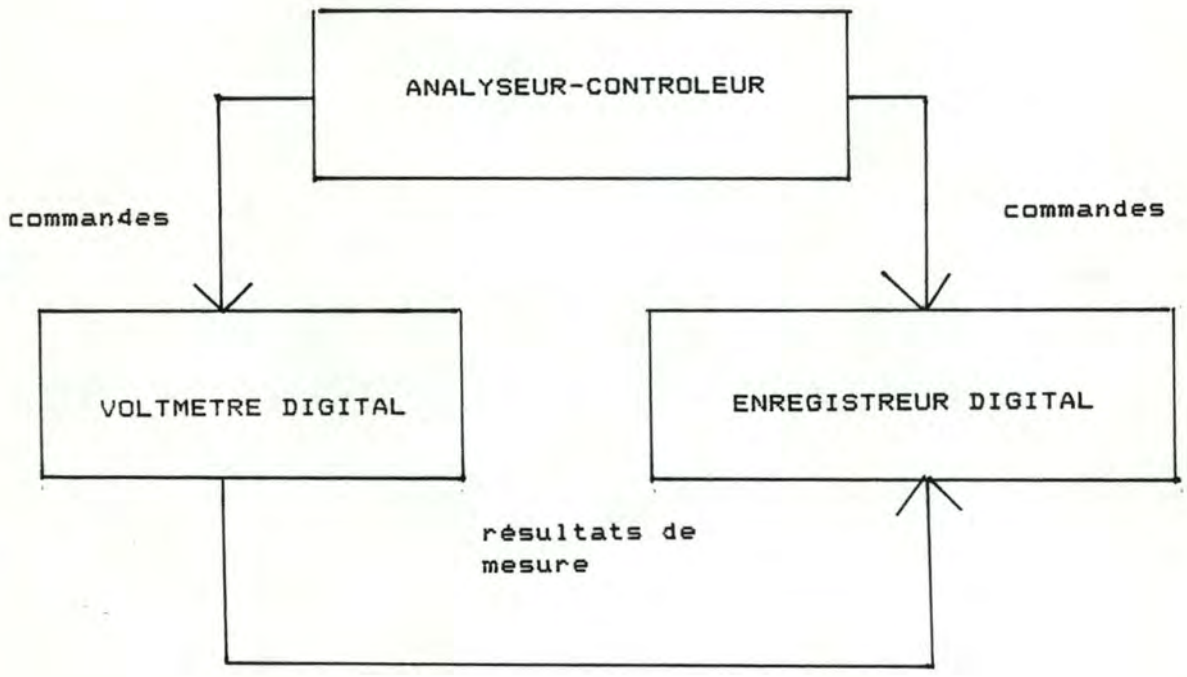


Fig. 1.1.0.1. Configuration-système autonome et de taille réduite.

Afin de résoudre les problèmes posés par la conversion des codes et formats de transfert, des adaptateurs furent développés pour réaliser la connexion de nouveaux appareils au sein de la configuration.

L' évolution conjointe de la technique informatique et de l' électronique de laboratoire vers la fin des années 60, donna lieu à un accroissement sensible des volumes de données à transférer, ainsi que des taux de transfert.

L' apparition, sans cesse croissante, d'instruments intégrant le micro-processeur, ainsi que d' autres circuits LSI, permit d' uniformiser la circuiterie d' interfacage, ainsi que les procédures de programmation des instruments.

Tout en conservant la même structure en étoile, les nouvelles configuration-systèmes introduirent progressivement le concept de transfert parallèle et bidirectionnel.

Vers le début des années 70, l' architecture modulaire de certains mini-ordinateurs incita les constructeurs à entreprendre le développement de nouvelles configurations à partir d'une structure de bus unique.

Cette nouvelle philosophie permit de normaliser les codes et formats des données, d' accroitre les taux de transfert, de réaliser une communication bidirectionelle et de déléguer le contrôle à plusieurs appareils.

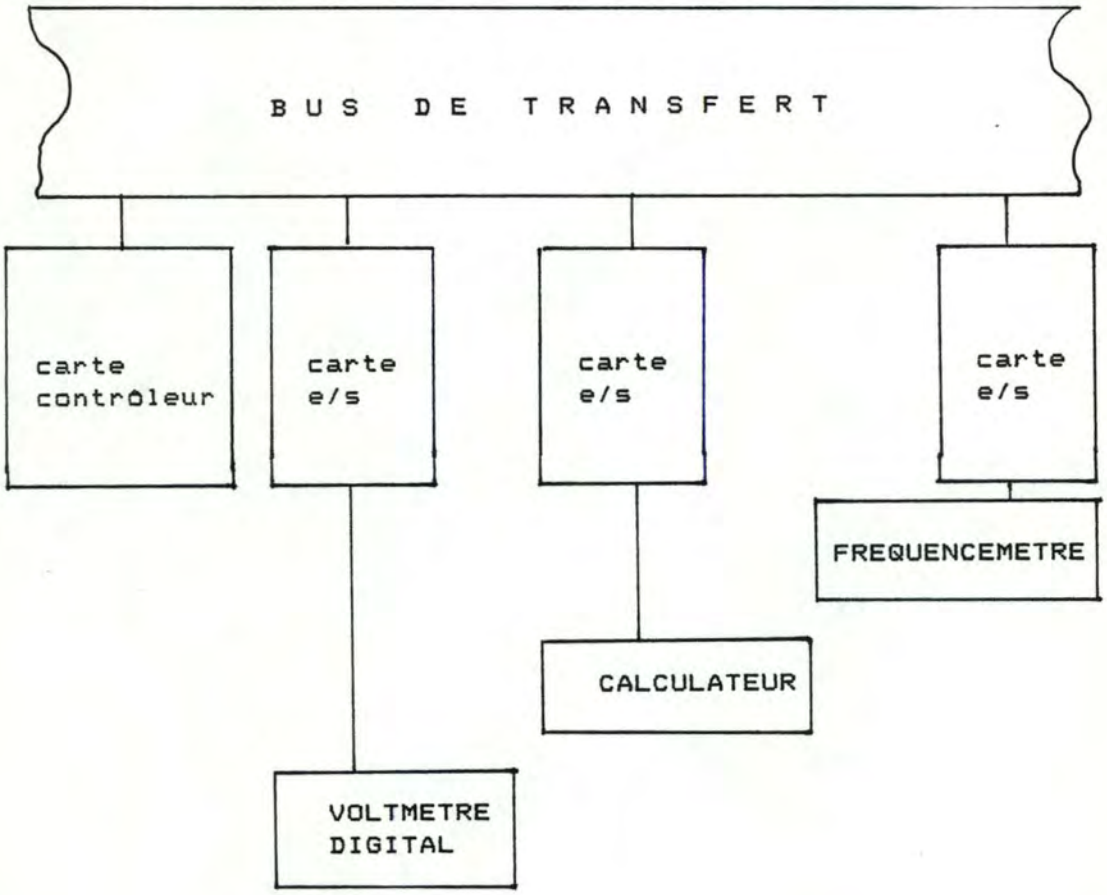


Fig. 1.1.0.2. Architecture à bus unique.

Du point de vue du bus, tous les appareils respectent les conventions d' interfacages grâce à la présence des cartes E/S d' adaptation.

Les appareils conservent cependant leurs propres caractéristiques fonctionnelles et ne sont donc pas liés à un type de bus donné. Un changement de configuration implique cependant un changement de cartes d' adaptation au nouveau bus, sans toutefois nécessiter une modification des appareils.

1.1.1. Le bus IEEE-488.

Le bus d'interfacage IEEE-488 s'inscrit dans la ligne de conduite du bus unique.

A la différence du dernier type d' architecture rencontré, les fonctions d' interfacage qui étaient jusqu'ici réalisées au moyen de cartes d' E/S d' adaptation, sont à présent supportées par chaque appareil de la configuration.

La connexion entre instruments s' éffectue à partir d'un câble de liaison, au même titre que le bus en fond de panier réalise la liaison entre les cartes d' adaptation d' E/S.

L'interconnexion d'une variété quelconque d' appareils consiste désormais à brancher le câble aux appareils à mettre en relation, sans aucune autre forme d' adaptation. Le câble de liaison possède une vocation passive dans le transfert et ne constitue à ce titre qu'un élément externe au système.

L' originalité du système de bus réside dans sa souplesse de transfert du flux d'information : grâce à l' interconnexion en parallèle des différents éléments de la configuration, un appareil peut addresser un ou plusieurs autres, sans nécessiter pour autant l' intervention d'un ordinateur, et l' introduction d'un organe de contrôle à l'intérieur du système à été largement simplifiée.

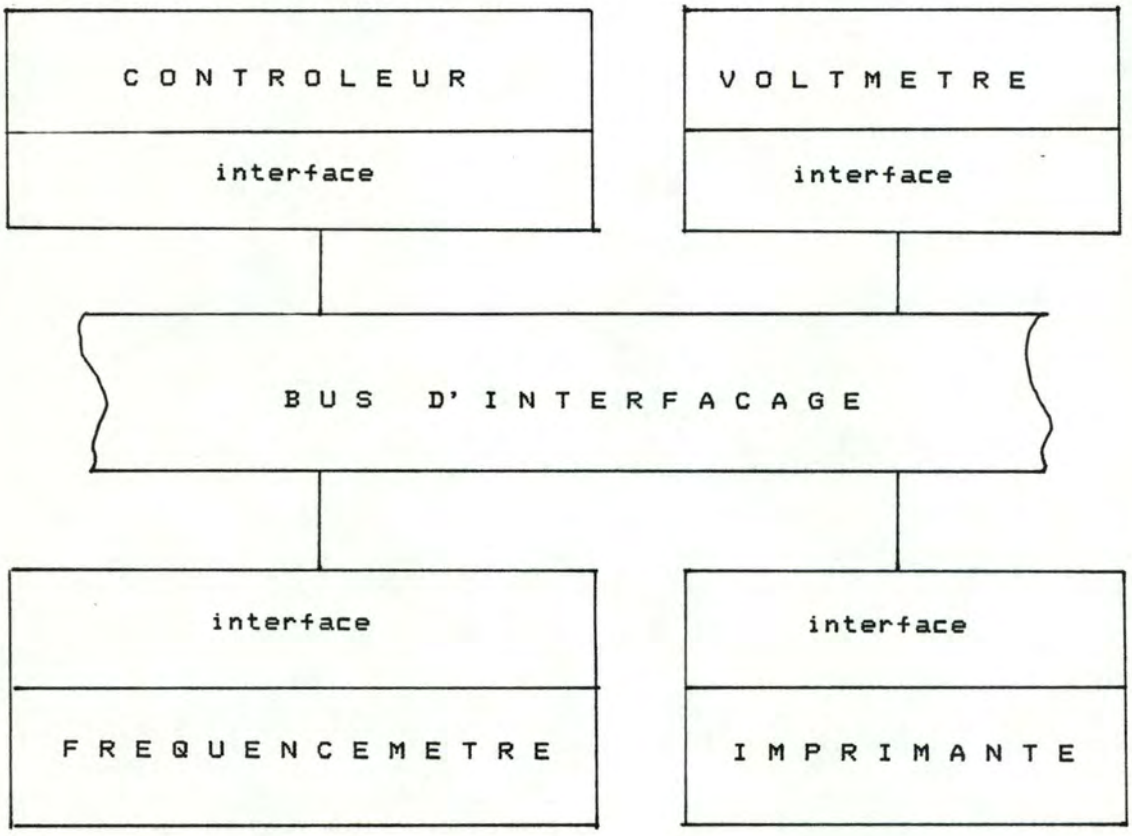


Fig. 1.1.1.0. Architecture à bus unique intégrant les fonctions d' interfacage au sein de chaque appareil.

Ce nouveau schéma d' interfacage offre, par ailleurs, la possibilité d' incorporer en un minimum de temps et en l' absence de toute connexion particulière, un nouvel appareil au sein de la configuration en place, sans aucune opération de reconfiguration du système que nécessitent le plus souvent les configurations traditionnelles.

1.2. La norme IEEE-488.

De l' évolution des structures d' interfacage, est né un intéret marqué pour la définition et l' adoption d' une norme d' interfacage applicable aux systèmes d' instrumentation.

Ce besoin correspondait principalement à la volonté de réaliser des connexions entre appareils issus de constructeurs différents, au prix d'un minimum, voire de l' absence, d' adaptation.

Les démarches qui furent alors menées permirent d' aboutir à la définition de la norme IEEE-488, telle que nous la connaissons aujourd' hui.

1.2.0. Historique de la norme.

Les premiers contacts relatifs à l' étude des objectifs et de la faisabilité d' une norme à portée internationale, débutèrent, tant aux Etats Unis qu' en Europe, vers le milieu de l' année 1972.

Ces réunions eurent pour principal objectif de tenter de trouver une réponse aux développements de systèmes d' instrumentation, qui proposaient, jusqu' alors, une solution distincte aux problèmes liés à l' interconnexion d' appareils au sein de configurations.

Le besoin de définir une norme d' interfacage pouvant être appliquée à un large éventail de produits et devant séduire par la même occasion tant les constructeurs que les utilisateurs de ces types d' appareils, se manifesta alors avec acuité.

Un groupe de travail, créé à l' initiative du Comité National Allemand et mené par la Commission Electrotechnique Internationale (I.E.C.), vit le jour dans le courant de l' année 1972 et élabora un premier projet de développement. Pareilles démarches furent menées parallèlement aux Etats Unis à la demande de constructeurs de matériel et d' utilisateurs. Les premiers concepts proposés par le constructeur américain Hewlett-Packard, pour à l'interfacage d' appareils de mesure, servirent alors de modèle et de base de discussion quant aux développements ultérieurs.

La première conférence tenue à Munich en 1972 par le groupe de travail allemand, permit de sélectionner, après une procédure d' approbation, la proposition américaine comme base d' élaboration de la norme.

Durant les deux années qui suivirent, de nombreux concepts de base relatifs à l' interfacage furent élaborés au niveau international et, en 1974, le bureau de normalisation I.E.E.E. marqua son accord pour l' approbation de la norme I.E.E.E-488.

Depuis lors, des efforts de clarification relatifs aux définitions de la norme, notamment par l' adoption d' une représentation graphique au moyen de diagrammes d' états, furent entrepris et donnèrent lieu à une révision de la norme en novembre 1978.

Ces changements et additions à la norme publiée en 1975, eurent pour principal effet d' éclaircir certains aspects de définition de la norme, sans cependant apporter de changements d' un point de vue technique.

Il fallut toutefois attendre 1980 pour voir publier la norme européenne I.E.C. 625-1 sous le titre "An Interface System for Programmable Measuring Appartus (Byte Serial, Bit Parallel)", correspondant en tout points, à l' exception du connecteur, à la norme I.E.E.E. 488.

L' adoption du connecteur de 25 broches par la norme européenne, en place du connecteur de 24 broches imposé par la norme I.E.E.E.-488, répond à un plus large usage qui est fait de ce type de connecteur. Cependant, l' utilisation intensive de ce connecteur à d' autres fins, et notamment pour les communications de données en série de type V24/RS-232, qui met en oeuvre des tensions et des courants différents, pourrait susciter une confusion auprès d' utilisateurs non avertis, et causer une déteroriation du matériel connecté.

Cette caractéristique de la norme I.E.C. a incité nombre de constructeurs européens à choisir le connecteur américain; ce dernier étant actuellement utilisé à plus de nonante pour cent en Europe.

1.2.1. Evolution de la norme.

La norme I.E.E.E.-488 régit les aspects mécaniques, électriques et fonctionnels de l' interface qu' elle propose, dans la mesure ou ces spécifications ne sont pas liées à un appareil ou un système donné.

L' aspect opérationnel, inhérent à tout système d' interfacage et qui concerne, notamment, le contenu et le format des messages, est laissé à l' initiative du concepteur de l' appareil, et ne donne lieu à aucune définition explicite de la norme.

Il appert qu' aucune convention unique de codification ou de format ne puisse être retenue dans l' élaboration d' une norme ayant, comme principal objectif, l' interconnexion d' un large éventail de produits, possédant des caractéristiques fonctionnelles spécifiques.

Cependant, au vu des efforts menés par certains constructeurs d' appareils, en vue de généraliser la présentation des messages, l' institut I.E.E.E. décida, tout récemment, de publier des recommandations relatives à la codification et au format des messages, sous la dénomination " I.E.E.E. Recommended Practice for Code and Format Conventions " (I.E.E.E. Std 728-1982).

Ce document définit un ensemble limité de lignes de conduites et d' alternatives, sous la forme d' un éventail de formats et de codes différents, susceptibles d' être supportés par l' ensemble des appareils intégrant l' interface I.E.E.E.-488. Ces conventions s' appuient, par ailleurs, sur l' usage d' une codification largement généralisée, qu' adoptent de nombreux constructeurs d' appareils intégrant l' interface.

1.2.2. Les objectifs du bus I.E.E.E.-488.

Suite au succès que connu un grand constructeur d' appareils de mesure en commercialisant une gamme d' appareils échangeant de l' information numérique au moyen d' une interface normalisée, il fut publié pour la première fois, en avril 1975, le texte de la norme I.E.E.E.-488 sous la dénomination " Digital interface for programmable instrumentation " .

Par la publication de ce texte, de nombreux constructeurs d' appareils numériques, qui se trouvaient jusqu' alors confrontés à un problème d' interconnexions entre appareils, bénéficiaient, par ce biais, d' une solution universellement reconnue quant aux problèmes d' interfacage rencontrés.

Des problèmes d' interprétation de certaines clauses de la norme ont nécessité une révision du texte aboutissant, en 1978, au texte connu sous le vocable de I.E.E.E.-488.

1.2.2.0. Les objectifs généraux du bus.

L' étude des textes permet de déceler les objectifs et les tendances suivantes :

- la simplicité :

Cet aspect est du, en grande partie, au succès rencontré par la norme auprès de nombreux constructeurs de matériel.

Le bus d' interfacage se compose de lignes électriques exclusivement passives, dont la circuiterie de fonctionnement se trouve intégrée à chacun des appareils mis en relation.

Cette forme d' agencement se caractérise, dès lors, par un faible coût de revient, nécessité habituellement par le développement d' appareils accessoires, tout en répondant d' une manière adéquate aux besoins de la configuration.

- l' universalité :

Les constructeurs d' appareils, offrant l' option I.E.E.E.-488, s' engagent à respecter scrupuleusement la norme.

Il reste cependant possible pour un constructeur, de n' offrir qu' un sous-ensemble des fonctions définies par la norme.

Une meilleure compatibilité entre systèmes est offerte lorsque certaines caractéristiques, bien que ne relevant pas du texte de la norme proprement dite, font cependant l' objet d' un consensus entre constructeurs. L' usage répandu du code A.S.C.I.I. comme moyen de codification de l' information constitue, à ce titre, un exemple d' entente implicite adopté par plusieurs constructeurs.

- la souplesse d' utilisation :

Cette caractéristique s' affirme davantage en distinguant par catégorie fonctionelle, les appareils concernés par un échange d' informations, et notamment :

- les organes de saisie de l' information.

Cette catégorie regroupe les appareils dont la vocation principale consiste à prendre en considération l' information quand elle se présente. Il s' agit le plus souvent d' appareils de mesure, de capteurs, d' analyseurs ou de tout autre type d' appareil analogue.

Habituellement, ces appareils proposent leurs propres conventions d' interfacage, et, la mise en oeuvre de plusieurs de ces appareils au sein d' une même configuration, suscitent le plus souvent un temps d' intégration relativement important.

L' incorporation de l' interface normalisée I.E.E.E.-488 permet de résoudre le problème de connexion physique ainsi que de conventions de transfert entre ces différents appareils.

- les organes de contrôle :

Il s' agit pour ces appareils d' orchestrer les transferts entre les appareils fonctionnant au sein d' une même configuration de système.

L' approche classique postulait la conception et le développement d' un contrôleur tenant compte des particularités des différents appareils constituants la configuration, débouchant sur une solution le plus souvent originale et mal adaptée à une extension possible de cette configuration, ultérieurement.

L' introduction d' un organe de contrôle normalisé apporte une plus grande souplesse dans la mise en oeuvre d' une configuration d' appareils. Outre l' avantage de la normalisation des procédures de contrôle, cette souplesse se caractérise principalement par la capacité d' extension de ce type de configuration, lors de l' introduction d' organes supplémentaires, notamment.

- les organes de diffusion des résultats.

Il s' agit, le plus souvent, d' appareils appelés à restituer l' information à destination du monde extérieur.

Cette catégorie comprend principalement les imprimantes, ainsi que les organes de visualisation des résultats.

La souplesse d' utilisation se caractérise principalement par la présentation, le plus souvent uniformisée, des résultats, ainsi que l' unicité des connexions. De plus, le caractère unidirectionnel de ces appareils, en ce qui concerne le transfert des données, permet au concepteur de concevoir un appareil ne comportant qu' un sous-ensemble restreint des fonctions d' interfacage définies par la norme.

- la souplesse de développement.

La norme I.E.E.E.-488 définit les caractéristiques fonctionnelles, électriques et mécaniques relatives au bus d' interfacage.

Cette faculté permet au concepteur de systèmes, de ne se soucier que des aspects relatifs à l' application proprement dite, sans devoir faire référence aux problèmes que pose habituellement le développement de l' interface.

1.2. Les objectifs formels du bus.

D' un point de vue plus formel, la norme s' attache à :

1.- définir un système de portée générale, susceptible d' opérer à l' intérieur d' un même site.

2.- spécifier, indépendamment du matériel, les interfaces de type mécanique, électrique et fonctionnel destinées à etre supportées par les appareils, afin d' assurer une parfaite communication à l' intérieur du système.

3.- spécifier la terminologie et les définitions associées au système.

4.- permettre l' interconnexion d' appareils issus de constructeurs différents, au sein d' une même configuration d' appareils.

5.- permettre une connexion directe entre appareils de complexités diverses.

6.- permettre une communication entre appareils, sans l' introduction d' un organe externe.

7.- minimiser les restrictions relatives aux performances globales de la configuration et du bus.

8.- définir un système réalisant les transferts selon un mode asynchrone, à partir d' un large éventail des taux de transfert.

9.- concevoir un système à faible coût, permettant la connexion d' appareils à bon marché.

La popularité de la norme a permis son adoption par d' autres institutions de normalisation, et, notamment :

- l' A.N.S.I. (American National Standards Institute) , A.N.S.I. MC1.1 " Digital Interface for Programmable Instrumentation " .

- l' I.E.C. (International Electrical Commission) , I.E.C. 625-1 " An Interface System for Programmable Measuring Appartus (Byte serial, Bit parralel) " .

Seule la norme européenne I.E.C. 625-1 diffère des deux autres en matière d' interconnexions.

Le texte définit l' utilisation d' un connecteur de 25 broches de type D. Cependant, ce type de connecteur est largement utilisé à d' autres fins, et notamment par la norme C.C.I.T.T. - V24 (RS232-C) qui met en oeuvre des tensions et des courants incompatibles avec les signaux de l' interface concerné.

Cette caractéristique à incité la majorité des constructeurs européens à adopter le connecteur prescrit par la norme I.E.E.E.-488 / A.N.S.I. MC1.1. plutôt que le connecteur proposé par l' I.E.C. .

Il convient finalement de mentionner que la norme I.E.E.E.-488 est largement diffusée sous d' autres dénominations qu' adoptent certains constructeurs de matériel, et, notamment :

* G.P.I.B (General Purpose Interface Bus),

* HP-IB (Hewlett-Packard Interface Bus),

* I.E.E.E. BUS ,

* PLUS BUS ,

* ASCII BUS .

1.2.3. Tendances Actuelles et Futures du bus I.E.E.E. 488.

La gamme et le nombre de produits incorporant l' interface I.E.E.E. 488 ne cessent de croître.

L' éventail comprend tant les appareils de mesure électroniques (voltmètres digitaux, fréquencemètres digitaux, analyseurs de spectre radio-fréquence), que les appareils d' équipements de laboratoires médicaux, les micro-ordinateurs domestiques et professionnels, les contrôleurs de mini-ordinateurs à grande échelle, les imprimantes et certains autres appareils de périphérie de systèmes.

Plus d' un million de produits intégrent aujourd'hui l' interface, et il est possible de considérer que le nombre de ces produits double en volume tous les deux ans, approximativement.

La majeure partie de ces appareils intégrent l' interface à titre principal (voire exclusif) à l' intérieur d' appareils de bancs de tests (appareils de mesure ou de simulation). Ce marché représente quelques 56 pour cent des produits vendus. Les contrôleurs prennent 11 pour cent du marché, alors que les organes de visualisation et de mémorisation n' en prennent que 8. La partie restante est constituée de systèmes complets de tout types.

Un large éventail de produits liés à l' interface I.E.E.E.-488 consiste en connecteurs, câbles, circuits conducteurs de lignes spécifiques, ainsi que de nombreux circuits L.S.I. qui réalisent les fonctions de contrôle et de gestion du transfert.

D' autres produits consistent, encore, en outils de maintenance ou appareils accessoires permettant une plus grande souplesse dans l' utilisation du bus. Citons à ce propos les appareils de visualisation de l' activité du bus, d' analyse d' états et de temps de réponse, ainsi que les instruments destinés à étendre l' utilisation du bus, par le biais d' une conversion du mode de transfert et permettant le raccordement au bus, en dehors des limites de distance imposées par la norme.

1.3. Anatomie du bus I.E.E.E.-488.

1.3.0. Apercu du bus d' interfacage.

Le bus d' interfacage dispose de seize lignes électriques, susceptibles de supporter un maximum de quinze appareils. Ces lignes constituent le seul lien physique entre les différents appareils, disposés chacun en parallèle sur les lignes de bus.

Fonctionnellement, le bus se partitionne en bus de transfert, bus de contrôle et bus de gestion du transfert.

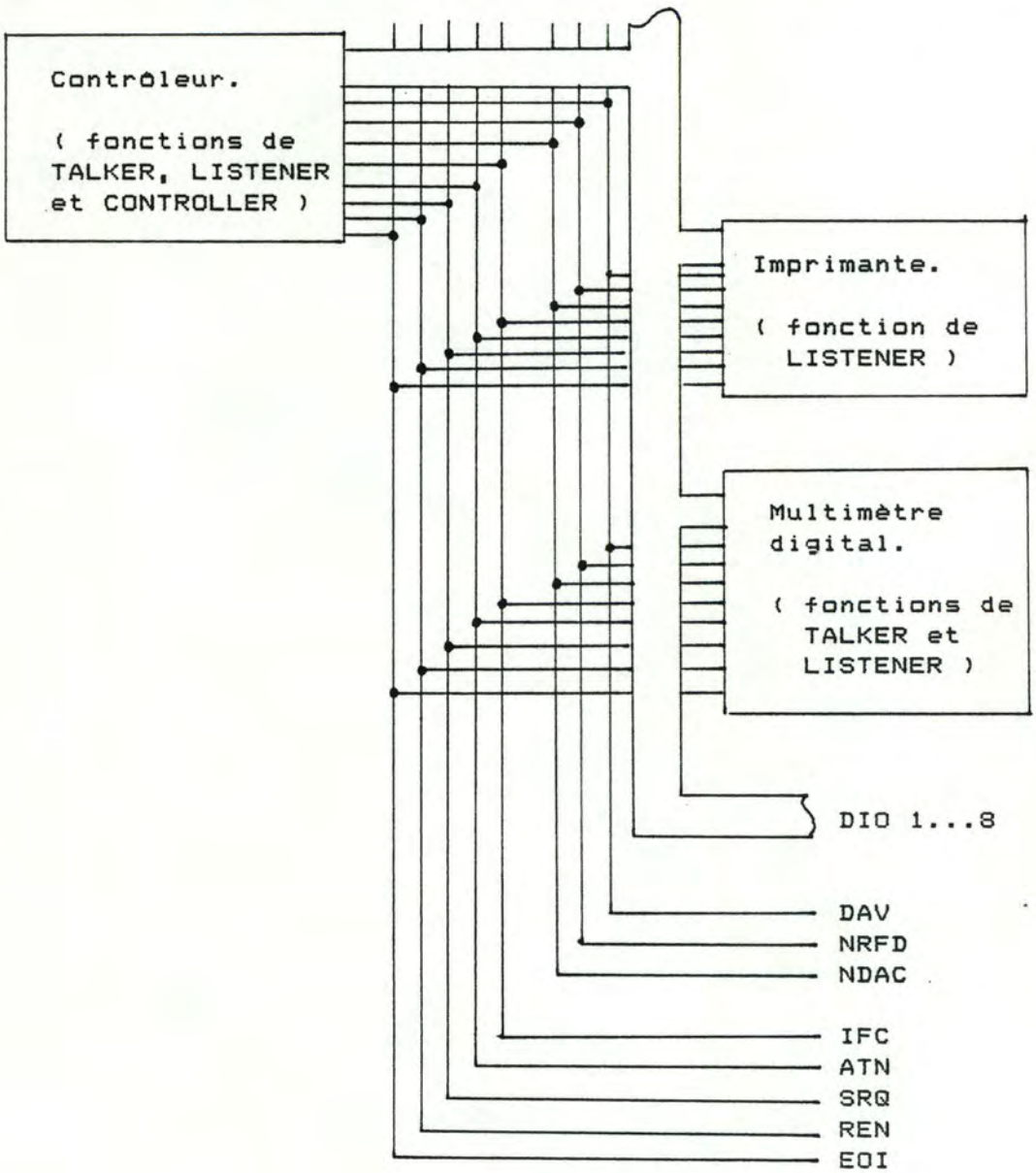


Fig. 1.3.0.0. Structure du bus I.E.E.E.-488.

Afin de pouvoir réaliser un échange de données, tout appareil doit pouvoir supporter, au moins, l' une des trois fonctions suivantes :

- la fonction TALKER : Cette fonction est supportée par un appareil susceptible de transmettre des informations à destination de un ou plusieurs autres appareils de la configuration. Des appareils dont la vocation principale est l' émission de données, tels les appareils de mesure, certains organes de restitution de l' information (enregistreurs digitaux,..), supportent cette fonction. Un seul appareil ne peut supporter la fonction de TALKER à un moment donné.

- le fonction LISTENER : Cette fonction est réalisée par un appareil susceptible de recevoir des informations d' autres appareils supportant la fonction TALKER, tout transfert ne pouvant invoquer qu' un seul TALKER à la fois. Des appareils tels des imprimantes, des instruments programmables, des organes de saisie de l' information, supportent cette fonction. Leur nombre est limité à quatorze pour une configuration de système.

- la fonction CONTROLLER : Cette fonction permet à un appareil de contrôler les échanges d' information réalisés sur le bus, en dédicacant les appareils susceptibles de pouvoir réaliser ces transferts. Un seul contrôleur ne peut être actif à un moment donné et, dans une configuration comprenant plusieurs contrôleurs, il ne peut y avoir qu' un seul contrôleur-système.

De nombreux appareils supportent plusieurs de ces fonctions. Ainsi, un multimètre fonctionne en LISTENER lorsqu' il recoit des commandes de configuration, et en TALKER pour l' émission de résultats.

Les transmissions réalisées sur le bus d' interfacage sont de nature bidirectionnelle. Il importe de signifier avant toute opération de transfert, les rôles impartis aux différents appareils appelés à réaliser ces transferts.

Cette signification est réalisée par un appareil supportant la fonction CONTROLLER qui déterminera "qui" fera "quoi", "quand" et "comment".

Cette fonction de contrôle est réalisée grâce à la faculté dont dispose le contrôleur, d' adresser les appareils appelés à converser, en leur assignant les fonctions de TALKER ou de LISTENER, suivant les cas.

Une fois dédicacés, les appareils réalisent le transfert sous le contrôle, mais sans l' intervention, du contrôleur. Ce dernier ne reprenant le contrôle du bus, qu' à la fin du transfert.

1.3.1. Description des lignes du bus d' interfacage.

Ces lignes sont au nombre de seize, et peuvent être répertoriées sous trois catégories distinctes :

1.3.1.1. Les lignes de gestion ou de contrôle du bus.

Afin de pouvoir gérer adéquatement le transfert entre plusieurs appareils de la configuration, le contrôleur dispose de plusieurs lignes de contrôle, posseédant chacune une signification distincte.

La ligne ATN.

Cette ligne relève de l' initiative exclusive, dont dispose le contrôleur, d' intervenir à tout moment sur le bus d' interfacage, aux fins de contrôle.

Une fois portée au potentiel électrique bas (signification logique vraie), les appareils présents dans la configuration entrent dans un " mode de commande " (COMMAND MODE), et se mettent à l' écoute du contrôleur.

Les instructions relatives à ce mode, et issues exclusivement du contrôleur, comprennent :

- les adresses de TALKER (talker address).
- les adresses de LISTENER (listener address).
- les commandes universelles (universal commands).
- les commandes d' adressage (addressed commands).
- les commandes de dé-adressage (unaddressed commands) .

En portant la ligne ATN au potentiel électrique haut (signification logique fausse), le contrôleur autorise, aux appareils préalablement dédicacés en mode de commande, l' accès aux lignes de données (DATA MODE).

Les appareils n' ayant pas été dédicacés en mode de commande, restent isolés du bus de données et ne peuvent en aucune facon participer au transfert.

La ligne REN.

En portant cette ligne au potentiel électrique bas (signification logique vraie), le contrôleur place les appareils supportant la fonction REMOTE, dans un état (REMOTE MODE) leur permettant de recevoir les commandes du contrôleur de facon exclusive (inhibition des commandes recues depuis un

clavier ou un panneau de contrôle de l' appareil).

Cette commande n' est opérée par le contrôleur, qu' après avoir adressé, au préalable, les appareils susceptibles d' être placés dans cet état (adressage en mode LISTENER).

Il incombe alors aux appareils ainsi adressés, de s' assurer de la présence du signal REN, et de l' interpréter endéans une période de 100 µsec.

Une fois portée au potentiel électrique haut (signification logique fausse), les appareils placés dans cet état retournent en mode LOCAL.

Cette faculté leur permet de recevoir à nouveau les commandes introduites à partir d' un clavier, ou de tout autre organe de contrôle associé à ces appareils.

La ligne IFC.

Le contrôleur possède la faculté d' interrompre toute opération en cours d' exécution sur le bus d' interfacage, en portant cette ligne au potentiel électrique bas (signification logique vraie).

Il incombe alors à tous les appareils de la configuration, de s' assurer de la présence du signal IFC, et de l' interpréter endéans une période de 100 µsec.

La ligne SRQ.

Cette ligne peut être utilisée à tout moment par un ou plusieurs appareils nécessitant l' attention particulière du contrôleur.

Cette attention est habituellement sollicitée lors de la communication par un appareil au contrôleur, de la présence de données à transmettre, ou de la détection d' erreurs (erreurs de syntaxe, de dépassement, d' activation trop rapide, etc...).

Le contrôleur dispose de la faculté de ne pas prendre en compte l' interruption (masquage du SRQ). La ligne conservera, alors, son potentiel électrique bas, signifiant que l' appareil solliciteur attend de voir honorée sa demande (service du SRQ).

Dans le cas de la prise en compte de l' interruption par le contrôleur, il s' agit de déterminer l' appareil (ou les appareils) à l' origine de la requête.

Le contrôleur dispose alors de la possibilité de mener une recherche par scrutation (POLLING). La norme prévoit une scrutation selon deux modes distincts : le mode série (SERIAL POLL), et le mode parallèle (PARALLEL POLL).

Ces deux modes seront analysés par la suite.

La ligne EOI.

L' utilisation conjointe du message ATN et EOI, permet au contrôleur de mener une interrogation en mode parallèle, du ou des appareils à l' origine de l' interruption (SRQ).

En l' absence de ATN, le signal EOI est utilisé par un appareil TALKER, lors d' un transfert de données, afin de signifier au contrôleur la fin du transfert en cours.

1.3.1.2. Les lignes de synchronisation.

Tout échange de données sur le bus d' interfacage, se réalise au moyen de trois lignes de synchronisation (handshakking lines), reproduites ci-dessous :

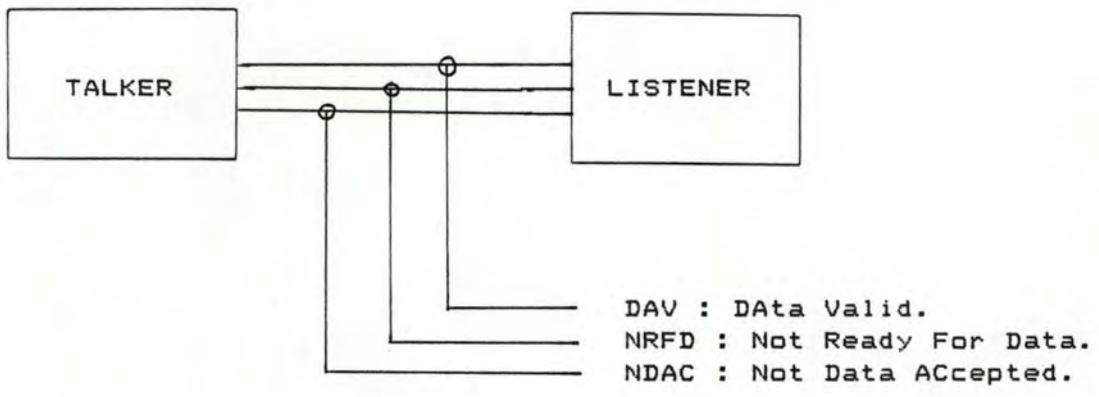


Fig. 1.3.1.0. Les lignes de synchronisation du bus I.E.E.E.-488.

Les lignes NRFD et NDAC constituent des lignes de contrôle de transfert à l' usage des appareils LISTENER de la configuration.

La ligne DAV est gérée par l' appareil TALKER, et marque la disponibilité d' un octet d' information sur le bus de données.

Ces lignes assurent l' intégrité du transfert, et offrent les avantages suivants:

- le transfert de données est asynchrone par nature. L' émetteur réalise le transfert à la vitesse dictée par l' appareil le plus lent de la configuration.

- la synchronisation est assurée au niveau de chaque byte émis.

- plusieurs appareils peuvent recevoir, au même moment, des données en provenance d' une source d' émission commune.

L' organigramme de la séquence de synchronisation entre parties participant à un échange de données est reproduite ci-dessous :

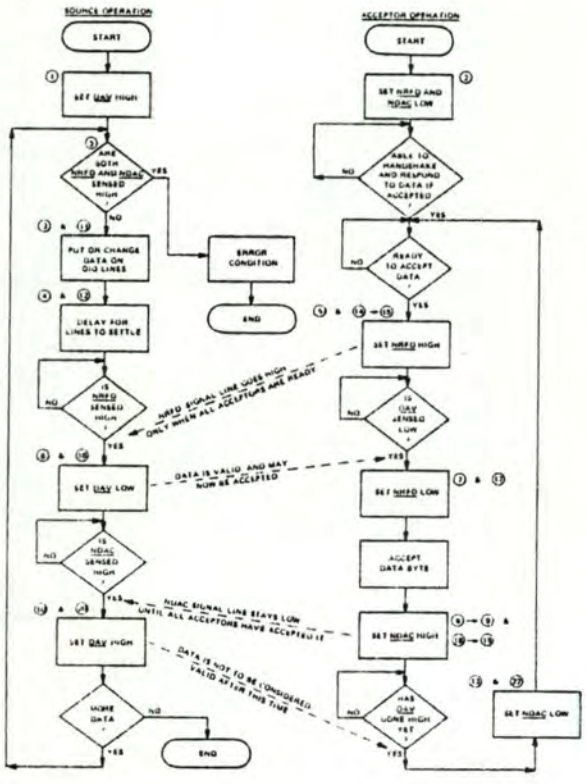


Fig. 1.3.1.1. Organigramme de la séquence de synchronisation entre les fonctions d' émission et réception de données.

Le transfert d' un mot sur le bus de données, est controlé par les lignes de synchronisation, à partir du schéma suivant :

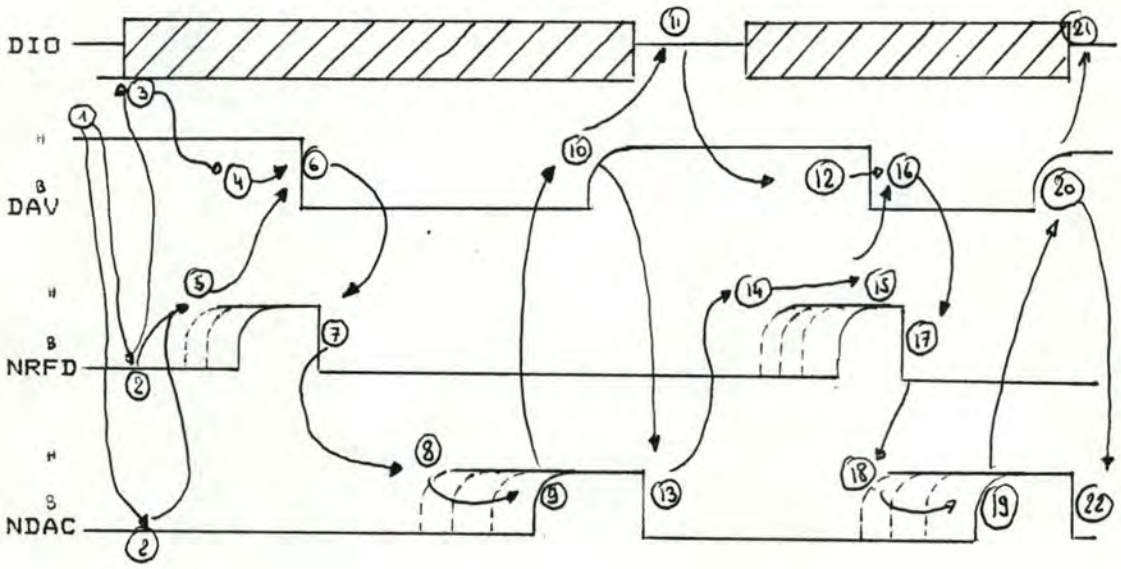


Fig. 1.3.1.2. Chronogramme des signaux de synchronisation.

Préalablement à tout transfert, l' émetteur s' assure de la disponibilité des appareils récepteurs à recevoir des données, en scrutant la ligne NRFD (Not Ready For Data).

En portant cette ligne au potentiel électrique haut, les appareils récepteurs signalent à l' émetteur leur état de disponibilité respectifs. Il est à noter que cet état de disponibilité ne sera éffectivement reflété, que lorsque le dernier appareil récepteur (et, par conséquent, le plus lent), se sera déclaré prêt pour le transfert. L' émetteur dépose alors sur le bus de données, le mot d' information et signale sa validité en portant la ligne DAV (Data Valid) au potentiel électrique bas (signification logique vraie).

Les récepteurs acceptent l' information à leur rythme propre, en positionnant dans un premier temps la ligne NRFD à un niveau électrique bas, marquant leur indisponibilité à recevoir de nouvelles données, et dans un second temps la ligne NDAC (Not Data Accepted) à un niveau haut, signifiant à l' émetteur l' acceptation de la donnée présente sur le bus.

Cette ligne ne prendra éffectivement ce dernier état, qu' après que l' appareil le plus lent de la configuration ait accepté la donnée. Après la prise en compte de l' information par tous les écouteurs, l' émetteur porte la ligne DAV à l' état haut, signifiant l' invalidité de l' information présente sur le bus de données.

En réponse, les écouteurs portent la ligne NDAC à l' état bas, afin d' initialiser la séquence de transfert suivante. Le nouveau transfert ne pourra avoir lieu, qu' après que tous les écouteurs aient marqué leur nouvel état de disponibilité à recevoir une nouvelle donnée, en positionant la ligne NRFD à l' état haut.

1.3.1.3. Les lignes de données.

Le bus d' interfacage dispose de huit lignes électriques (D101-D108) assurant un transfert bidirectionnel des données, réalisé selon une procédure asynchrone.

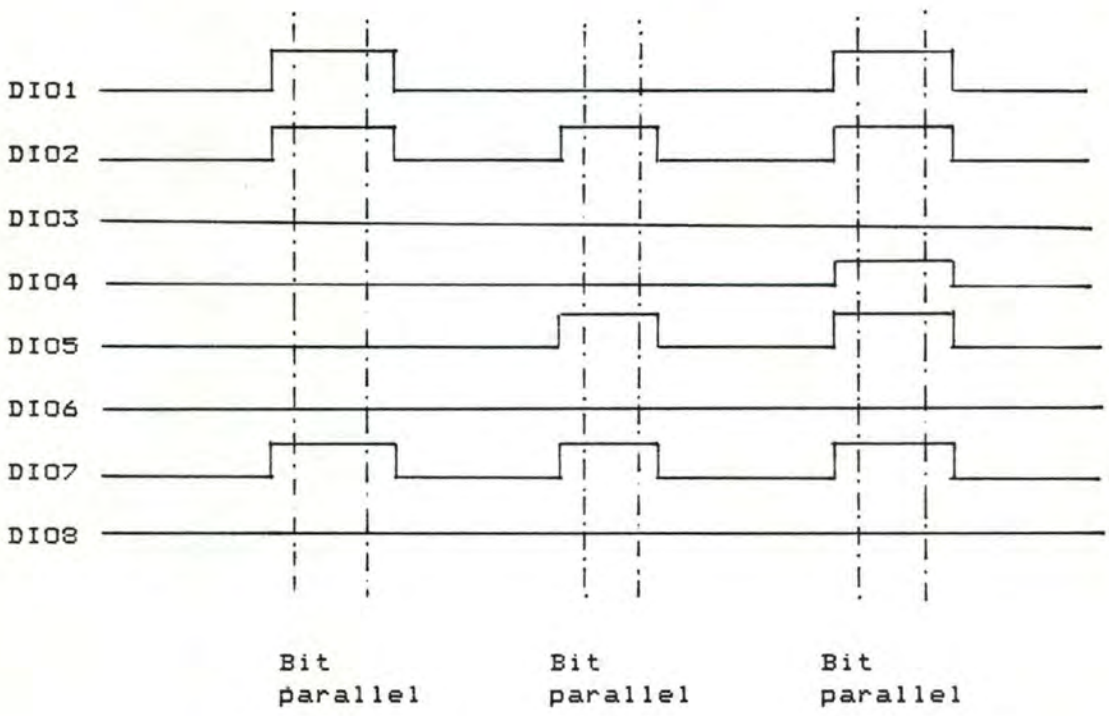


Fig. 1.3.1.3. Séquence de transfert.

Le transfert correspond à l' envoi en parallèle (procédé dit BIT PARALLEL), et en simultanéité, de huit signaux électriques constituant un mot d' information (BYTE).

Chaque mot constitutif du transfert est alors transmis en série au rythme des signaux de synchronisation (procédé dit BYTE SERIAL).

Bien qu' habituellement les codes A.S.C.I.I et I.S.O soient les plus couramment utilisés en matière de transferts, tout autre convention de codification de l' information peut, toutefois, être retenue.

Seules les informations transmises en mode de commande, et concernant notamment la fonction d' adressage, sont soumises au respect de la norme en matière de contenu.

1.3.2. Analyse des types de données.

La nature de l' information véhiculée sur le bus de transfert, est étroitement liée aux deux modes d' échanges d' information que supporte le bus, et déterminée par l' état de la ligne de contrôle ATN.

La ligne ATN portée au potentiel électrique haut (signification logique fausse), signifie que les informations qui circulent sur le bus, sont des données ou des informations d' état, et sont, par conséquent, étroitement liées à la nature de l' application rencontrée.

En portant la ligne ATN à l' état électrique bas, le contrôleur signifie que l' information présente sur le bus, doit être interprétée comme des adresses d' appareils ou des commandes. Ces informations sont reprises sous la dénomination de "mode de commande", par opposition aux informations du "mode de données" rencontrées plus haut.

1.3.2.0. Les informations du mode de données.

Ces informations sont transmises sur le bus de transfert, avec la ligne de contrôle ATN, portée au potentiel électrique haut.

L' information qui est alors transmise est habituellement liée à l' application même, et consiste le plus souvent en résultats de mesure, communication d' informations d' états, données de mémorisation, paramètres de configuration ou de contrôle destinés à la partie fonctionnelle d' un appareil, etc...

En raison du caractère par trop spécifique de l' information véhiculée, aucune règle de format ou de contenu de cette information n' est reprise dans la norme.

Cette information est véhiculée par l' interface, à destination de la partie fonctionnelle de l' appareil, et ne possède, à ce titre, aucune autre forme de relation avec la partie d' interfacage de l' appareil.

1.3.2.1. Les informations du mode de commande.

Ces informations sont transmises sur le bus de transfert, avec la ligne de contrôle ATN, portée au potentiel électrique bas.

Tous les appareils se comportent alors comme des LISTENERS et scrutent le bus de données, afin de reconnaître la commande, dans un premier temps, pour l' exécuter ensuite, dans le cas ou cette commande leur était destinée.

Il convient d' opérer ici une distinction entre ces commandes, transmises sur le bus de transfert avec la ligne ATN portée à l' état électrique bas, des commandes issues des lignes de contrôle du transfert, que nous rencontrerons plus loin, dans notre analyse des types de données.

Les commandes d' adressage TALKER et LISTENER.

Ces commandes font l' objet d' un adressage sur le bus de transfert, et ont pour effet de dédicacer les appareils en mode TALKER ou LISTENER selon le rôle qui leur est assigné par le contrôleur, préalablement au transfert.

Le choix du mode de transfert (TALK ou LISTEN) se fait par un adressage spécifique, réalisé à l' intérieur de l' une des deux classes d' adresses, reprises dans le tableau ci-dessous :

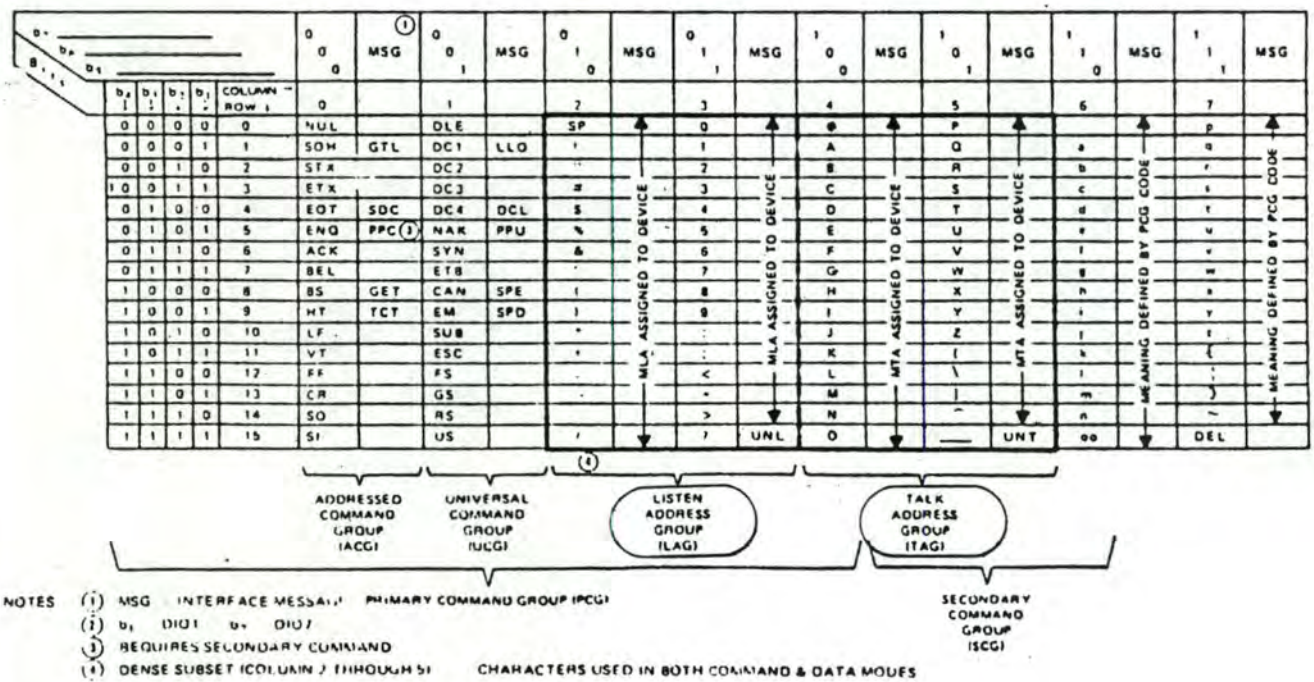


Fig. 1.3.2.0. Tableau des adresses de TALKER et LISTENER.

L' envoi d' une adresse de TALKER sur le bus de transfert, en mode de commande, aura pour effet de dédicacer un instrument en lui assignant un rôle d' émetteur dans le transfert, dès la reconnaissance par l' appareil de son adresse propre. De même, l' émission de une ou plusieurs adresses LISTENER a pour effet d' assigner un rôle de récepteur aux appareils ainsi adressés.

Il convient cependant de remarquer que seul l' adressage en mode LISTENER peut être réalisé à destination de plusieurs appareils. L' adressage d' un TALKER a pour effet de désadresser tout autre appareil émetteur de la configuration, conformément au principe voulant qu' un seul émetteur ne soit actif à un moment donné.

Chacune des adresses que l' on désigne sous le vocable de "primaire", peut faire l' objet d' un adressage étendu (adressage secondaire), permettant une extension au concept de l' adressage primaire, qui limite le nombre d' appareils pouvant être adressés à 31.

Cette extension de l' adressage initial permet, par ailleurs, de pouvoir configurer un ensemble d' appareils selon un certain ordre logique de disposition au sein de la configuration, en adoptant, par exemple, une même adresse primaire pour les appareils appartenant à un même type fonctionnel, et une adresse secondaire distincte pour chaque appareil de cette catégorie.

Il convient de remarquer que seules les adresses primaires sont régies par la norme et doivent, à ce titre, respecter les contraintes de codes et de formats qui leur sont imposées. De même, deux ou plusieurs appareils peuvent se voir assignés la même adresse. Les transferts réalisés à destination de cette adresse seront alors communs aux deux appareils, sans possibilité de différenciation de la part de l' organe de contrôle.

Les commandes multilignes.

Ces commandes font l' objet d' une description fonctionnelle précise dans la norme, et se partitionnent en :

- commandes universelles.

Ces commandes sont envoyées à l' initiative du contrôleur, à l' ensemble des appareils présents dans la configuration, afin de permettre la réalisation d' une opération d' interfacage spécifique.

Les cinq commandes multilignes sont :

la commande Device Clear (DCL).

Cette commande permet de placer tous les appareils supportant cette fonction dans un état déterminé, généralement relatif au fonctionnement même de l' appareil.

la commande Local Lockout (LLO).

Cette commande inhibe toute commande en provenance d' un panneau avant, ou tout autre commande permettant de reprendre le contrôle de l' appareil depuis ce dernier. Cette commande ne fait pas l' objet d' un adressage, et doit être reconnue par tout appareil étant en mesure de la supporter. Le positionnement de la ligne REN à l' état électrique haut (signification logique fausse), a pour effet de replacer l' appareil en mode local, et d' annuler ainsi l' effet initial de la commande.

la commande Serial Poll Enable (SPE).

Cette commande définit le mode de scrutation des interruptions en série. Après un adressage préalable en émission, chaque appareil sollicité envoie un mot d' état. La prise en compte de cette commande suppose l' existence de la fonction TALKER dans le chef de l' appareil, afin de pouvoir émettre le mot d' état.

la commande Serial Poll Disable (SPD).

Cette commande a pour effet de terminer l' opération de scrutation des appareils en mode série, en désadressant les appareils préalablement appelés à participer à ce mode de scrutation.

la commande Parallel Poll Unconfigure Command (PPU).

Cette commande inhibe toute possibilité de réponse à une scrutation menée en mode parallèle, par tout appareil supportant cette fonction.

- commandes adressées.

Les commandes adressées, bien que fonctionnellement semblables aux commandes universelles, ne s' appliquent qu' aux appareils préalablement adressés en mode LISTENER.

Ces commandes sont au nombre de cinq :

1. La commande Group Execute Trigger (GET).

Cette commande permet aux appareils supportant cette fonction, et pour autant qu' ils aient été préablement adressés comme tels, de réaliser une action déterminée, telle une activation de mesure, la prise en compte d' un résultat à un moment donné, etc... .

2. La commande Selected Device Clear (SDC).

Cette commande permet de placer des appareils ayant été adressés préalablement en mode LISTENER, dans un état d' initialisation, définis le plus souvent à partir des caractéristiques fonctionnelles de l' appareil même. Il s' agira, le plus souvent, d' un état associé à une mise sous tension de l' appareil.

Cette commande est fonctionnellement semblable à la commande universelle DCL.

3. La commande Go To Local (GTL).

La commande Go To Local à pour effet de replacer tous les appareils adressés en mode LISTENER, et se trouvant dans l' état REMOTE, dans l'état LOCAL, leur permettant de recevoir à nouveau les commandes depuis le panneau avant de l' appareil.

4. La commande Parallel Poll Configure (PPC).

Cette commande permet à tous les appareils adressés préalablement en LISTENER et supportant cette fonction, d' être configurés pour une gestion des interruptions en mode parallèle, lors de la réception de la commande secondaire correspondante.

5. La commande Take Control (TCT).

Cette commande est issue de la faculté que possède un contrôleur actif, de transférer le contrôle à un autre appareil de la configuration, pour autant que ce dernier supporte les nécessaires fonctions de contrôle.

- commandes secondaires.

Les commandes secondaires permettent d' étendre le concept d' adressage d' un octet, tel que rencontré jusqu' à présent, à un adressage sur deux octets, utilisé en mode de scrutation des interruptions en parallèle.

Ces commandes sont au nombre de deux :

1. La commande Parallel Poll Enable (PPE).

La commande PPE est complémentaire à la commande adressée PPC, et a pour effet de solliciter le positionnement par l' appareil addressé, d' une ligne d' état, parmi les huit lignes du bus de transfert.

2. La commande Parallel Poll Disable (PPD).

Cette commande a pour objet d' inhiber l' effet de la commande adressée PPC, permettant à un appareil de ràpondre à une scrutation des interruptions menée en mode parallèle.

1.4.0. Aspects fonctionnels de l' interface.

La norme préconise, dans un souci de formalisation du concept d' interfacage, le partitionnement d' un appareil selon trois volets distincts :

1. la présentation des messages sous forme codée.
2. les fonctions propres à l' appareil et liées à son mode de fonctionnement.
3. les fonctions d' interfacage, indépendantes du mode de fonctionnement de l' appareil.

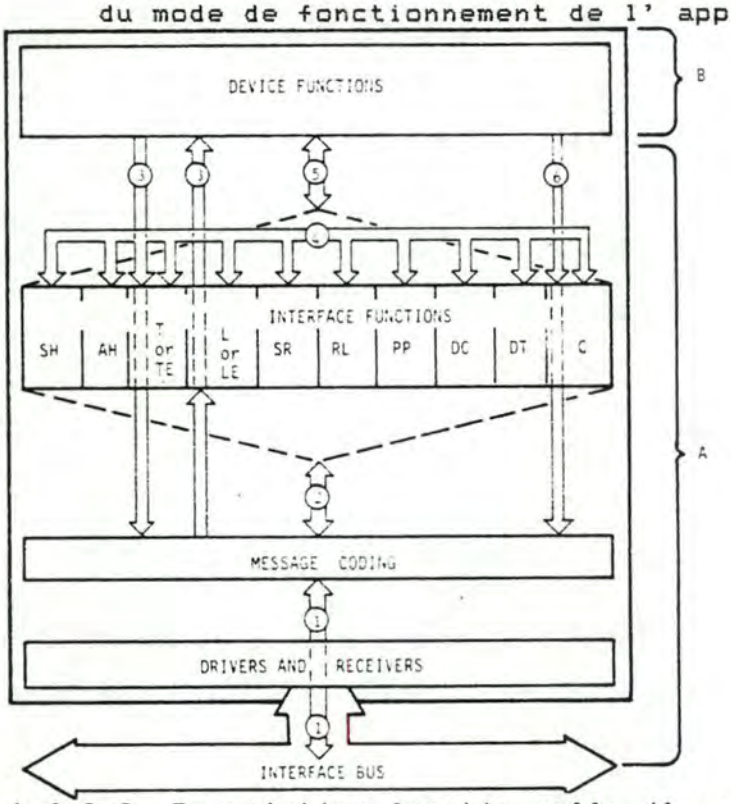


Fig. 1.4.0.0. Description fonctionnelle d' un appareil.

1.4.0.0. Le concept de message.

Les messages véhiculés sur les lignes du bus, relèvent de deux catégories distinctes : les messages liés à l' appareil et à son fonctionnement, et les messages liés aux fonctions d' interfacage de l' appareil, et par conséquent indépendantes de l' appareil proprement dit, ainsi que de son fonctionnement.

Alors que la première catégorie de messages ne relève pas de la norme, et ce, en raison du caractère par trop spécifique des fonctions supportées par l' appareil, la seconde catégorie fait l' objet d' une définition explicite de la part de

la norme, et évite, de ce fait, toute ambiguité dans son interprétation.

La distinction entre ces deux types de messages sera établie ci-après pour des raisons de commodités d' interprétation, sous les dénominations " messages liés au fonctionnement de l' appareil " et " messages non-liés au fonctionnement de l' appareil ".

1.4.0.0.0. Les messages liés au fonctionnement de l' appareil.

Ces messages, quoique ne relèvant pas de la norme, font cependant l' objet de recommandations en matière de format et de codification par l' institution I.E.E.E. . Le contenu même de ces messages est, toutefois, laissé à la discrétion du concepteur de l' application.

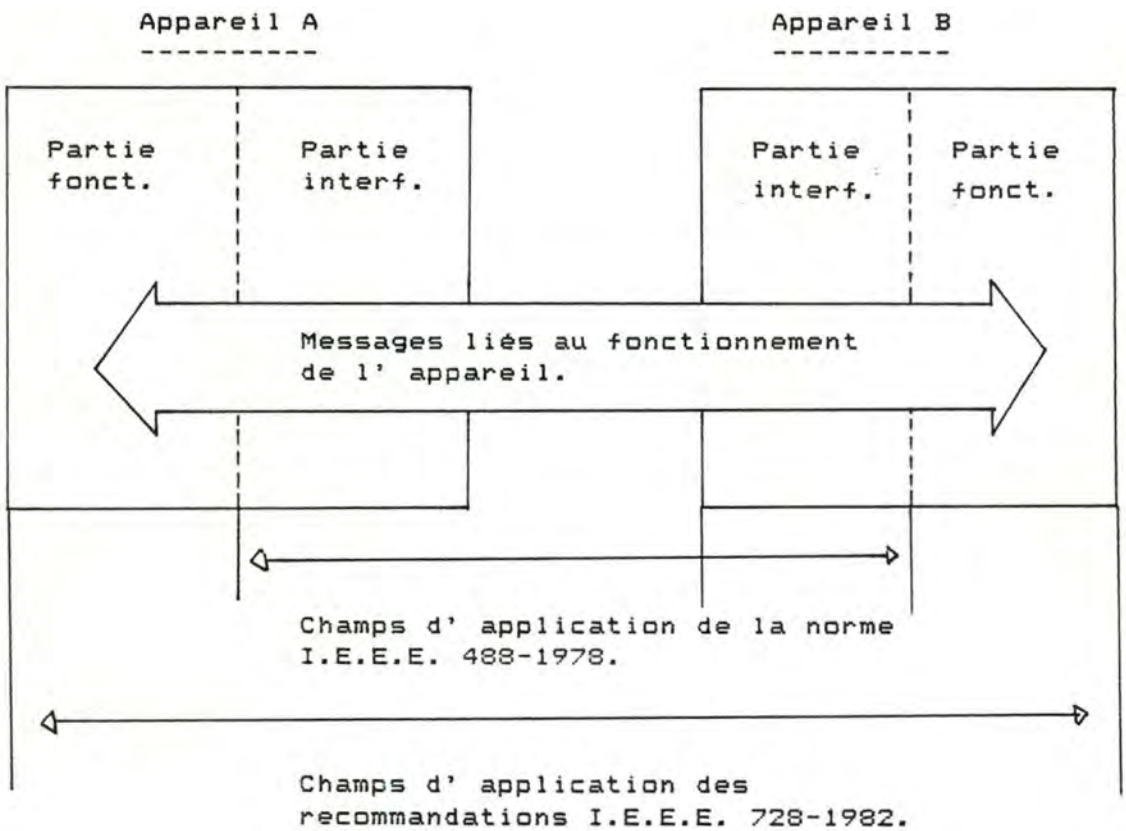


Fig. 1.4.0.1. Domaines d' application des messages liés au fonctionnement de l' appareil, et portée de la norme.

Un message est identifié, ici, comme une séquence d' octets véhiculés sur le bus de données. La compatibilité en matière de transfert de données est assurée grâce à une organisation et une structure commune, qu' adoptent les appareils concernés par l' échange.

Les recommandations de l' I.E.E.E. (I.E.E.E. Std 728-1982) préconisent de répertorier un message, lié au fonctionnement de l' appareil, sous quatre catégories distinctes :

1. les messages issus de résultats de mesure.

Ces messages consistent en informations obtenues en sortie d' un appareil, à la suite d' un processus de mesure.

Le contenu de ces messages est alors relatifs à des résultats de mesure pouvant s' exprimer sous forme de tension, de courant ou de fréquence, ou de tout autre unité de mesure dépendant de l' application.

2. les messages de programmation.

Un message de programmation consiste habituellement en une ou plusieurs séquences d' instructions destinées à initialiser ou exécuter une fonction de mesure, ou tout autre fonction supportée par l' appareil.

Cette procédure procède le plus souvent selon deux phases distinctes : la réception du programme et son exécution.

3. les messages d' informations d' états.

Une information d' état est relative aux conditions de fonctionnement interne d' un appareil. Cette information est le plus souvent communiquée à la demande, et peut prendre la forme d' indication d' erreur, de dépassement de mesure, etc...

4. les messages d' affichage.

Ces messages consistent en une information pertinente, transmise à destination d' un utilisateur, et relative à l' opération en cours. Il peut s' agir de résultats de mesure ou d' instructions relatives à l' utilisation de l' appareil.

Cette information donne habituellement lieu à une interprétation de la part de l' utilisateur.

Afin de satisfaire les recommandations établies par le document I.E.E.E. Std 728-1982, tout message doit pouvoir s' insérer dans une de ces quatres catégories de messages, avant de pouvoir répondre aux contraintes de format dictée à chaque catégorie de message.

1.4.0.0.1. les messages n' étant pas liés au fonctionnement de l' appareil.

Ces messages régissent les échanges entre une fonction d' interfacage et son environnement, menés sans référence à l' aspect fonctionnel de l' appareil.

Dans un souci de formalisation des procédures de transfert, la norme définit le message, comme une quantité d' information susceptible de véhiculer à tout moment une signification logique " vraie " ou " fausse ".

A l' intérieur de cette catégorie de messages, il convient d' opérer une distinction entre les messages dits " locaux " (Local Messages) et les messages distants (Remote Messages).

1.4.0.0.1.0. Les messages locaux (Local Messages).

Ces messages permettent l' échange d' informations entre les fonctions supportées par un appareil dans le cadre de son fonctionnement, et les fonctions d' interfacage relevant de la norme.

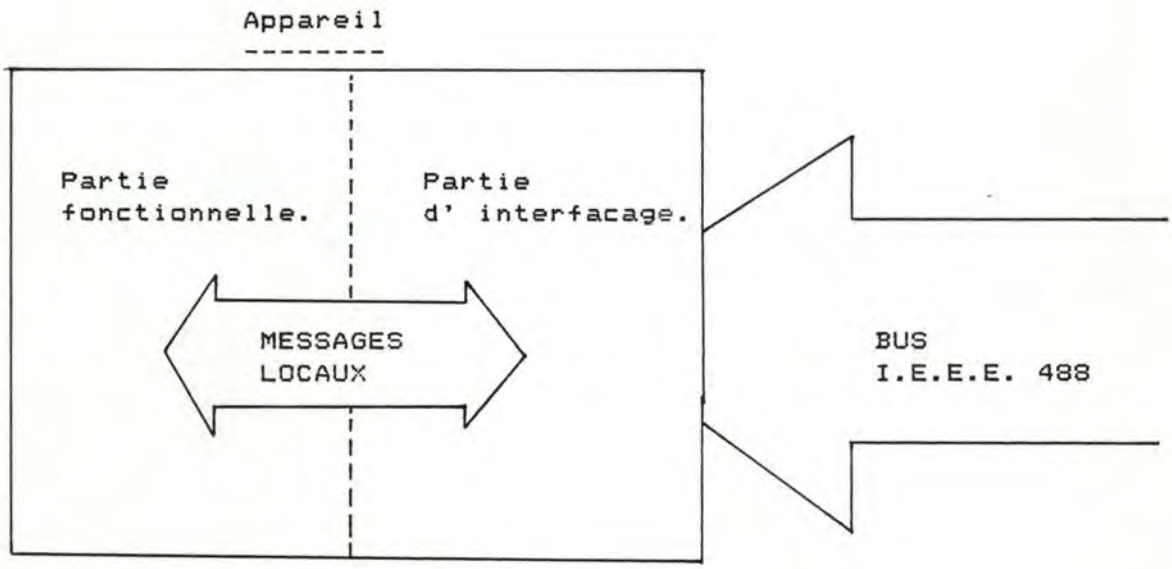


Fig. 1.4.0.2. Domaine d' application des messages locaux.

Les messages locaux sont transmis de facon interne à l' appareil, et n' ont aucun lien avec l' environnement extérieur.

Le concepteur ne peut, en aucun cas, définir un nouveau message, s' ajoutant à l' ensemble des fonctions d' interfacage. Tout au plus lui est-il loisible d' adjoindre un message local dérivé de n' importe quel état d' une fonction d' interfacage, à destination d' une ou plusieurs fonctions propres à l' appareil.

Afin de réaliser les nécessaires transitions entre états d' interfacage, la norme prévoit, dans le transfert des messages locaux propres à l' appareil, une durée minimale de présence de ces messages.

1.4.0.0.1.1. Les messages distants (Remote Messages).

Les messages distants, à l' opposé des messages locaux, sont véhiculés par le bus d' interfacage entre les différentes fonctions d' interfacage des appareils mis en présence.

Il s' agit, ici, des messages distants qui sont limités à la partie d' interfacage de l' appareil, et susceptibles de provoquer des transitions entre les états d' autres fonctions.

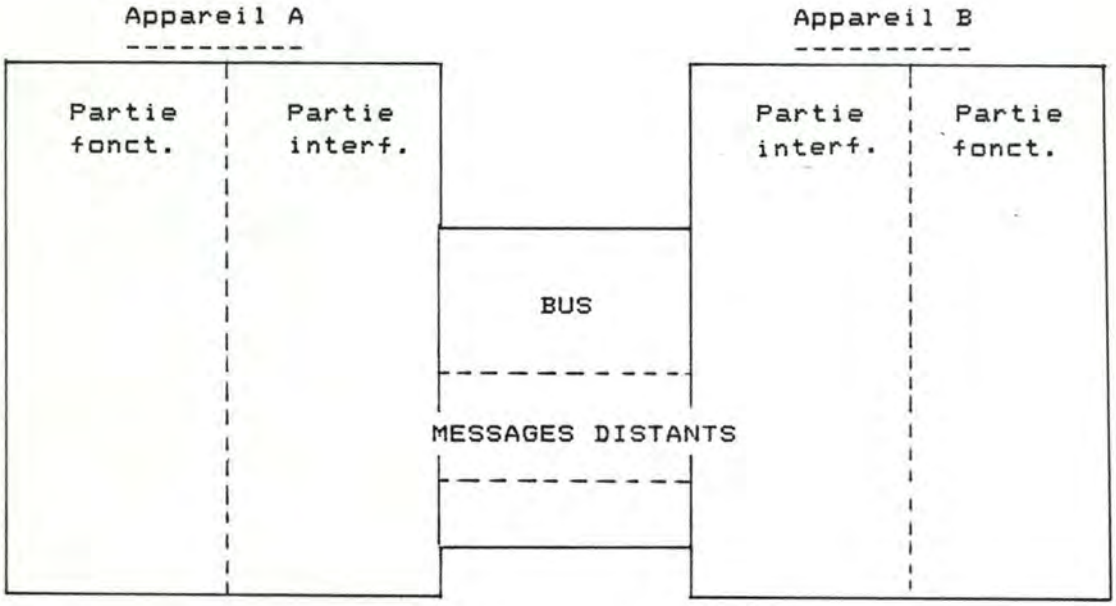


Fig. 1.4.0.3. Domaine d' application des messages distants.

Il convient d' opérer ici une distinction entre cette catégorie de messages distants, et les messages distants liés au fonctionnement même d' un appareil, évoqué dans le paragraphe précédent.

Cette dernière catégorie de messages, quoique véhiculés à travers les fonctions d' interfacage, ont cependant pour origine ou pour destination, la partie fonctionnelle de l' appareil, et sont, dès lors, de nature à agir sur cette composante de l' appareil.

Ces messages consistent le plus souvent en données de programmation, de paramètres opératoires, de résultats de mesures ou de communications d' informations d' états n' étant soumis à aucune prescription en matière de contenu ou de format par la norme. Seules certaines recommandations peuvent donner lieu à une certaine cohérence dans leur présentation et leur contenu.

1.4.0.0.2. Liens entre états.

Les transitions d' états associées à une fonction d' interfacage, sont issuent d' une interconnexion logique existant entre deux fonctions d' interfacage distinctes, par laquelle la transition vers un état actif de l' une des deux fonctions, est subordonnée à l' existence d' un état actif dans l' autre fonction d' interfacage.

Schématiquement, le lien entre ces états peut être représenté comme suit :

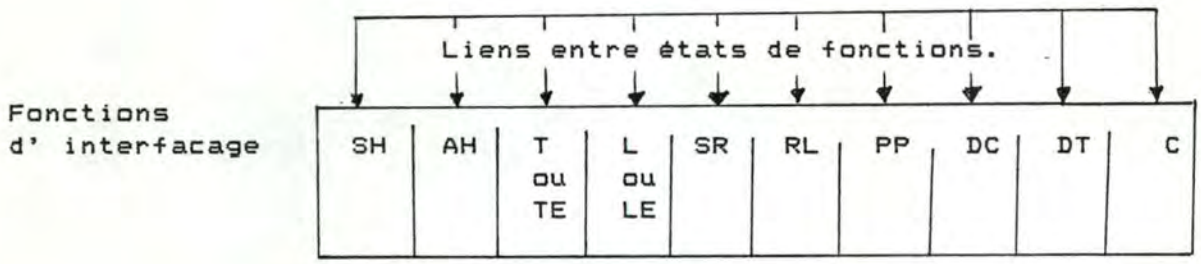


Fig. 1.4.0.4. Liens entre états de fonctions.

Chaque fonction est représentée au moyen d' un bloc fonctionnel, marquant le caractère d' imperméabilité de chacune des fonctions entre elles.

1.4.0.0.3. Codage des messages.

La procédure de codification s' applique aux messages distants, qui nécessitent une conversion en signaux électriques, afin de pouvoir être véhiculés sur le bus d' interfacage.

Certaines lignes supportent, à elles seules, un message distant. Ces messages sont alors appelés messages à ligne unique, par opposition aux messages à lignes multiples, pour lesquels la combinaison de plusieurs lignes fournit l' information.

Alors que plus d' un message à ligne unique peuvent être transmis simultanément, les messages à lignes multiples, utilisant le bus de transfert, ne peuvent être transmis qu ' à tour de rôle.

1.4.0.0.4. Les messages à lignes multiples.

Les messages à lignes multiples sont véhiculés sur les 8 lignes du bus de transfert, au même moment de la transmission du message à ligne unique, ATN.

A partir de ce moment précis, tout appareil présent dans la configuration, est en mesure de recevoir et traiter le message multiligne disponible sur le bus, pour autant qu' il supporte les fonctions qui lui sont associées.

L' inventaire des ces messages a été établi dans un chapitre précédent. Rappelons toutefois que ces messages consistent en :

1. Commandes Universelles (destinées à l' ensemble des appareils de la configuration).

2. Commandes adressées (destinées aux appareils faisant l' objet d' un adressage préalable).

3. Adresses Primaires (destinées à l' ensemble des appareils de la configuration).

4. Commandes ou Adresses Secondaires (destinées aux appareils ayant fait l' objet d' un adressage primaire ou d' une commande préalable).

1.4.0.1. Les fonctions de l' appareil.

Les fonctions associées à un appareil, dépendent à la fois de ses caractéristiques propres et de son mode de fonctionnement. Ces fonctions sont, à ce titre, étroitement liées, et, par conséquent, dépendantes de l' application rencontreé.

La norme n' a pas cru bon, pour ces raisons, de devoir définir des règles de fonctionnement ou des conventions de formats de messages de transfert dans ce domaine particulier, et laisse toute liberté au concepteur de l' appareil de pouvoir définir de nouvelles fonctions ou de nouveaux modes de fonctionnement, pour autant que ces derniers n' entrent pas en conflit avec la partie d' interfacage de l' appareil qui reste soumise à la normalisation.

1.4.0.2. les fonctions d' interfacage.

Ces fonctions sont indépendantes des caractéristiques fonctionnelles d' un appareil et ne sont, par conséquent, en rien liées à une application spécifique.

Une fonction d' interfacage constitue un élement du système, qui offre la faculté de receyoir, émettre ou traiter des messages.

Ces fonctions sont ménees à partir de conventions de transfert qui leur sont propres, et qui font l' objet d' une définition précise dans le texte normatif. Chacune des fonctions ne peut recevoir ou émettre un ensemble exhaustif de messages qu' à l' intérieur d' une classe de messages spécifiques.

1.4.0.2.0. Etats des fonctions d' interfacage.

Chaque fonction d' interfacage, peut se définir en terme de un ou plusieurs états en interrelation, mais mutuellement exclusifs. Un seul état ne peut être actif à un moment donné, et ceci quelque soit son degré d' interrelation avec d' autres états du système, au même moment.

Chaque état d' une fonction d' interfacage, associe les deux concepts suivants :

1. les messages susceptibles d' être véhiculés par l' interface, à l' intérieur d' un état donné.

2. les conditions de transition d' un état vers un autre, en liaison avec la fonction.

1.4.0.2.1. Conventions de représentation des fonctions d' interfacage.

1.4.0.2.1.0. Répresentation des fonctions au moyen de diagrammes d' état.

La représentation des fonctions d' interfacage, telles que formulées par la norme I.E.E.E.-488, se réalise au moyen de diagrammes d' état, qui répondent aux contraintes de formulation suivantes.

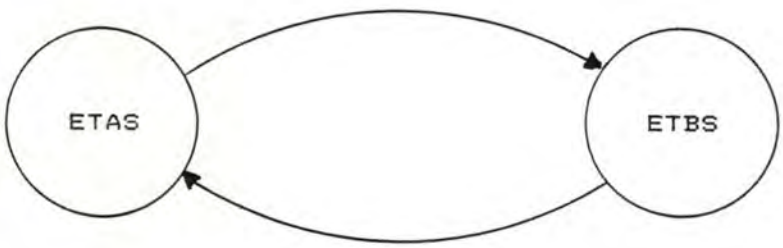
1. Représentation d' un état.

Chaque état associé à une fonction d' interfacage, se représente graphiquement au moyen d' un cercle.

L' état se définit au moyen d' un nom symbolique, constitué de quatre lettres majuscules, inscrites à l' intérieur du cercle. Le nom donné à l' état est défini par la norme, et se termine toujours par la lettre S (State) .

2. Représentation de la transition.

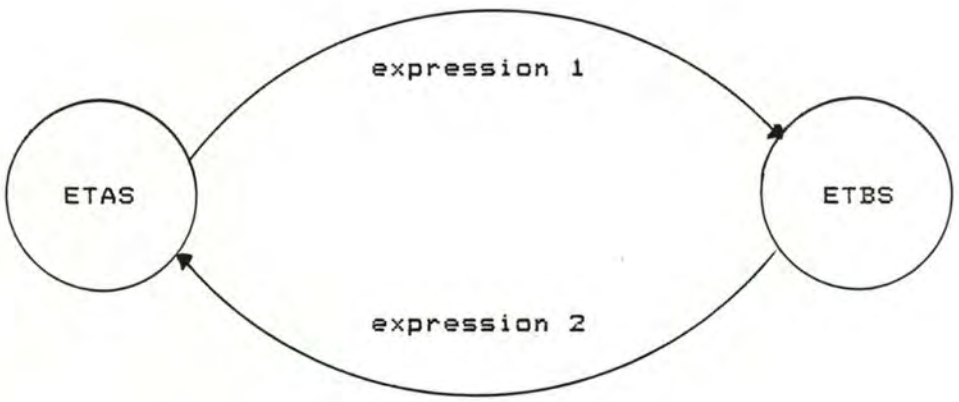
Toute transistion susceptible de se produire entre états, se représente au moyen d' arcs, qui renseignent sur le sens de la transition à opérer.



3. Expression de la transition.

Chaque transition se caractérise par une expression de type booléenne, dont la valeur s' exprime par l' assertion vraie ou fausse.

La transition d' un état vers un autre, ne s' éffectue, que lorsque l' expression associée à la transition se vérifie (assertion vraie). Dans le cas contraire, la transition ne s' éffectuera pas, et la fonction d' interfacage conservera le même état.



Une expression, consiste en un ou plusieurs messages locaux ou distants, de liens entre états, ou des limites de temps, auxquels peuvent être appliqués les opérateurs logiques ET, OU et NON.

4. Contraintes relatives aux expressions.

4.1. Expression de messages locaux et distants.

Afin de pouvoir dissocier les messages locaux des messages distants, la norme prescrit l' usage de lettres minuscules dans l' expression de messages locaux, et de lettres majuscules dans l' expression de messages distants.

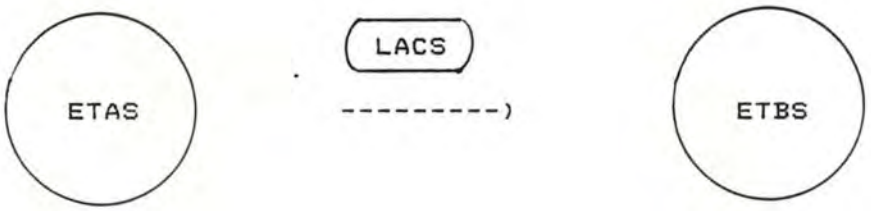
- exemple de formulation d' un message local de mise sous tension de l' appareil:

- exemple de formulation d' un message distant

4.2. Expression du lien issu d' un autre diagramme d' état.

L' expression du lien issu d' un autre diagramme d' état, se réalise au moyen du nom symbolique de l' état associé à la transition à opérer.

- exemple de formulation du lien issu d' un autre diagramme d' état :



L' expression de la liaison est vraie, lorsque l' état est actif au moment de l' évaluation. Dans le cas contraire, l' expression est fausse, et la transition ne s' opère pas.

4.3. Expression des contraintes de temps.

L' expression d' une durée de temps minimum, se formule au moyen du symbole Tn. Ce symbole ne prend la valeur "vraie", qu' après que l' interface ait conservé, durant une durée prescrite par la norme, le même état. Les valeurs associées à ces durées, sont reprises dans un tableau, figurant dans le texte normatif.

4.4. Expression de l' opérateur logique 'ET'.

L' opérateur logique 'ET' ('AND'), se formule au moyen du symbole 'V' renversé, à l' intérieur des expressions de la transition.

4.5. Expression de l' opérateur logique 'OU'.

La formulation de l' opérateur logique 'OU' se réalise au moyen du symbole 'V'.

Il convient de noter, que l' opérateur 'ET' est évalue avant l' operateur 'OU', en l' absence de parenthèses.

4.6. Expression de l' opérateur logique 'NON'.

L' opérateur logique 'NON', se représente au moyen d' une ligne horizontale, placée au dessus de l' expression ne se vérifiant pas.

4.7. Expression facultative.

Toute partie d' une expression, dont l' évaluation est facultative quant à la réalisation de la transition, se représente aux moyens de crochets ouverts et fermés.

4.8. Remarques générales.

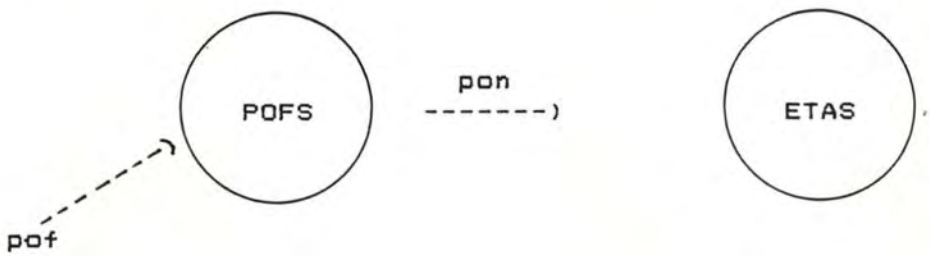
La formulation d' une expression de transition en provenance de tous les états d' un diagramme vers un même état de destination, peut se faire sous la forme abrégée suivante :

Cette forme de représentation évite la lourdeur de formulation que susciterait la procédure normale, et exprime la possibilité pour une expression de trouver son origine à partir de tout autre état.

cas particulier : La mise sous tension peut donner lieu à la forme de représentation abrégée, lorsqu' il n'est pas envisagé de tenir compte de l' état d' absence de tension comme pertinent.

La représentation, sous forme abrégée, se donne alors comme suit :

Dans le cas, ou l' état d' absence de tension doit être considéré comme étant pertinent dans le schéma de transition, la représentation suivante peut alors être adoptée:



1.4.0.2.1.1. Représentation des fonctions d' interfacage sous forme d' automates séquentiels.

Chaque fonction d' interfacage peut prendre la forme d' un automate séquentiel, utilisant les diagrammes d' état issus de la norme, et dont le mécanisme de fonctionement comprend:

1. Les entrées du système.

Celles-ci se présentent sous la forme de conditions primaires (C.I.), dont les origines sont :

- externes.

Il s' agit alors d' entrées issues du bus même, et qui comprennent :

- les lignes de synchronisation DAV, RFD et DAC, symbolisées par "bhs".

- les lignes de contrôle du bus (ATN, IFC, REN, EOI, SRQ), symbolisées par "bc".

- les messages de type multilignes, c' est à dire émis à partir du bus de données et symbolisés par "bio".

- internes.

Il s' agira, le plus souvent, de messages locaux, issus de la partie fonctionnelle de l' appareil, et symbolisés par "ml", ou, de l' existence d' autres états, existant à ce moment dans d' autres modules de l' interface. Ces derniers seront symbolisés par "eif".

- particulières.

Ces conditions sont issues de contraintes de temporisation qu' impose la norme. Elles seront représentées ici sous le symbole "T".

Ces conditions primaires apparaitront dans le graphe au moyen d' arcs orientés, évoquants la transition se produisant à partir d' un état d' origine (EP), vers un état cible (ES).

2. Les mémoires du système.

Ces mémoires prennent la forme d' états (E) dits secondaires, et qui correspondent à la définition d' états de la norme. Ceux-ci respecteront donc les contraintes de représentation déjà évoquées, et se présenteront sous la forme d' un sigle mnémonique entouré d' un cercle.

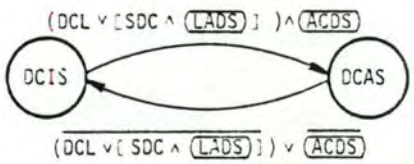
3. Les sorties du système.

Les sorties sont assimilables aux actions à entreprendre lors de chaque transition vers un nouvel état. Afin de ne pas surcharger le graphe, ces dernières seront présentées dans un tableau séparé.

A titre d' illustration, nous donnons ici la représentation de la fonction Device Clear (DC), sous cette forme.

Mémoires du sytème ou états (E) :

DC State Diagram



Entrées du système ou conditions primaires :

DC Mnemonics

Messages		Interface States	
m	DCL = device clear SDC = selected device clear	DCIS DCAS	= device clear idle state = device clear active state
eif	(ACDS) = accept data state (AH function) (LADS) = listener addressed state (L function)		

Sorties du système :

DC Message Outputs

DC State	Remote Message Sent	Device Function (DF) Interaction
DCIS	none	normal device function operation
DCAS	none	DF should return to a known fixed state

1.4.1. Aspects électriques de l' interface.

La norme définit de manière précise les caractéristiques électriques de l' interface, pour autant que les deux conditions suivantes soient remplies :

- les liaisons électriques sont limitées dans l' espace.

- le bruit électrique doit être de faible niveau.

1.4.1.0. Spécifications des niveaux électriques.

La définition des niveaux électriques en tension, se base sur l' utilisation généralisee de la technologie T.T.L. (Transistor to Transistor Logic), dans le développement de circuits électroniques.

La signification logique associée aux niveaux de tension, correspond aux limites suivantes :

Signification logique	Niveau en tension
0	> = 2.0 V (niveau électrique haut)
1	< = 0.8 V (niveau électrique bas)

Fig. 1.4.1.0. Limites des niveaux en tension et correspondances logiques.

Les limites des niveaux supérieurs et inférieurs correspondent aux limites fixées en technologie T.T.L., et établies respectivement à 5.25 V et au niveau neutre (masse).

Chaque conducteur de ligne doit , par ailleurs, être en mesure d' assurer une commutation en courant de l' ordre de 48 mA en permanence.

Afin de pouvoir préserver un niveau constant en tension lorsque tous les appareils se trouvent dans un état de haute impédance sur les lignes, chaque appareil sera pourvu d' une résistance de charge, ayant pour objet de maintenir une impédance de ligne constante et d' assurer une certaine immunité au bruit.

Configuration de circuits conducteurs de lignes.

Le schéma donné ci-dessous reprend, en guise d' illustration, la circuiterie d' interfacage au bus, ainsi que les caractéristiques des tensions, courants et composants mis en oeuvre.

R11: 3k ohmn à 5 pour cent (approx.)
R12: 6.2k ohm à 5 pour cent (approx.) et en réference à la masse.
Vcc: 5.25 Volts.

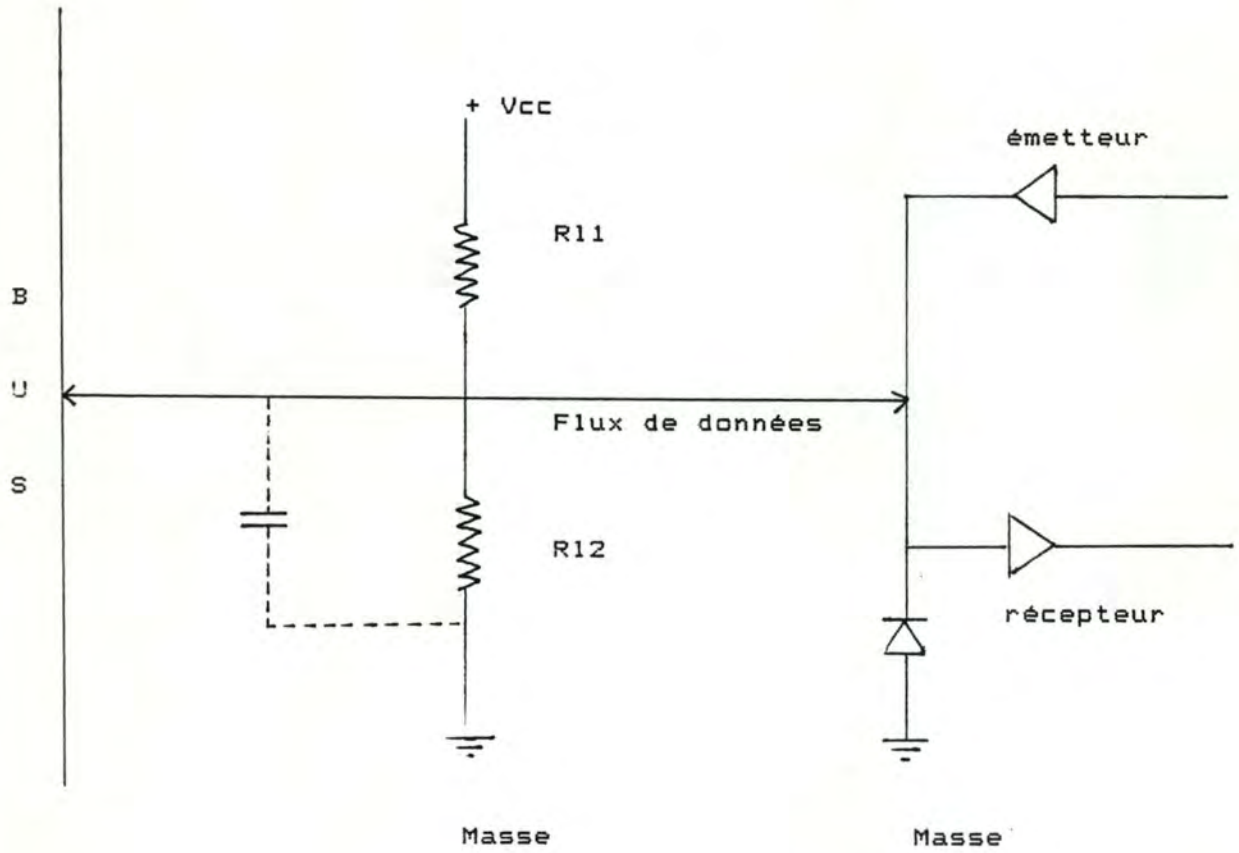


Fig. 1.4.1.1. Electronique d' interfacage aux lignes du bus.

1.4.1.1. Caractéristiques des conducteurs de lignes.

Tout message véhiculé sur le bus, peut être transmis de manière active ou passive.

Dans le cas d' une transmission passive, le transfert s' éffectue dans un état intermédiaire aux états binaires définis plus haut, et à partir de conducteurs de lignes disposants de collecteurs dits "ouverts".

Cette particularité s' applique notamment aux conducteurs des lignes SRQ, NRFD et NDAC, ce qui permet d' associer la gestion de ces lignes à un module logique de type "AND", tel qu' illustré par les lignes de synchronisation suivantes :

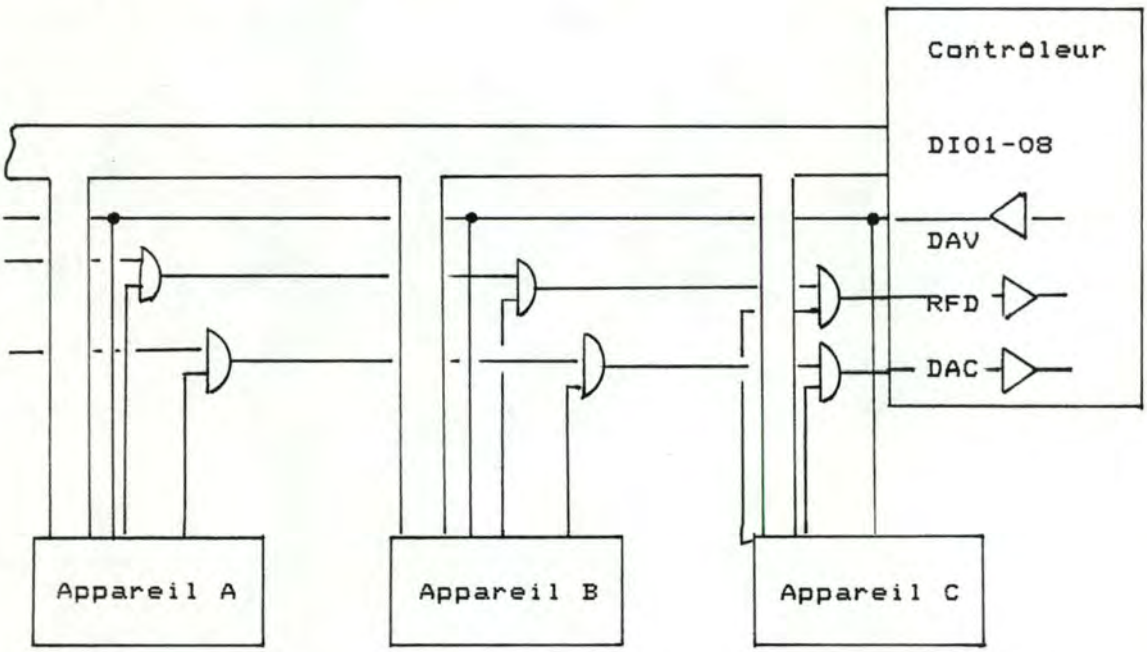


Fig. 1.4.1.2. Représentation symbolique de la connexion à collecteurs ouverts.

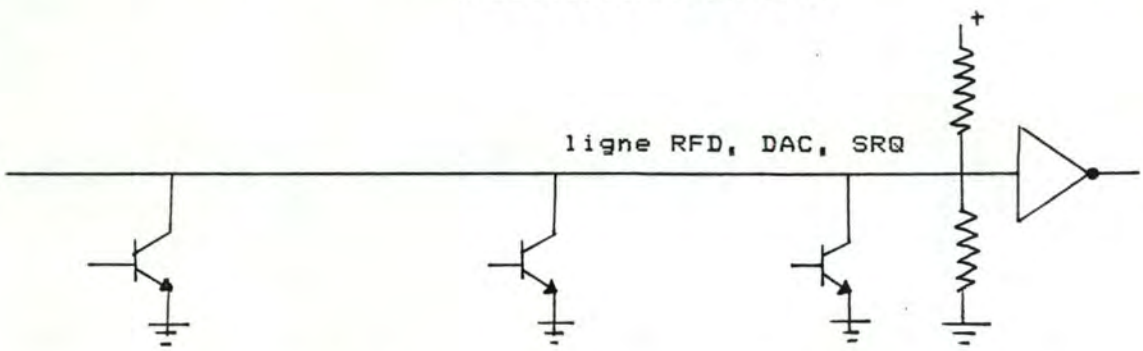


Fig. 1.4.1.3. Représentation électronique de la connexion à collecteurs ouverts.

Lorsque tous les conducteurs des lignes cités ci-avant, sont dans un état de "blocage", le niveau en tension des lignes est alors déterminé par les résistances R1 et R2 du schéma électronique (habituellement haut).

Il suffit alors qu' un seul conducteur de ligne commute ce niveau au potentiel bas, pour que la ligne reflète ce nouvel état.

Dans la séquence de synchronisation, il faut remarquer que :

- RFD, au niveau haut, signifie que tous les appareils sont prêts à accepter les données.

- RFD, au niveau bas, indique que l' un des appareils, au moins, n' est pas disposé à recevoir des données.

- DAC, au niveau haut, signifie que tous les appareils ont accepté les données présentes sur le bus de données ou de commandes. Cet état n' est vérifié qu' au moment ou l' appareil le plus lent de la configuration marqué cet accord, ce qui permet de réaliser une procédure de transfert asynchrone, et de tenir compte de la diversité des temps de réponse des appareils.

- DAC, au niveau bas, marque le fait que l' information présente sur le bus n' a pas été acceptée par un appareil, au moins.

Dans le même ordre d' idées, mais selon un cablage réalisé à partir d' un module "OU", la ligne SRQ, à l' état électrique bas, indique qu' un appareil au moins sollicite l' attention du contrôleur.

Une scrutation menée selon l' un des deux modes étudiés, permettra d' identifier le ou les appareils solliciteurs.

Des conducteurs de lignes à collecteurs ouverts peuvent être également adoptés pour les lignes DIO, DAV, IFC, ATN, REN et EOI, avec cependant la contrainte suivante : les lignes DIO1-8 devront utiliser des collecteurs ouverts, dans le cas ou l' appareil supporte les fonctions de scrutation des interruptions selon le mode parallele.

L' adoption de circuits à collecteurs ouverts offre par ailleurs de meilleurs performances en matière de taux de transfert.

1.4.2. Aspects mécaniques de l' interface.

La liaison des appareils supportants l' interface I.E.E.E.-488, se réalise par le biais de câbles et connecteurs normalisés, permettant un agencement des appareils selon une structure linéaire ou en étoile.

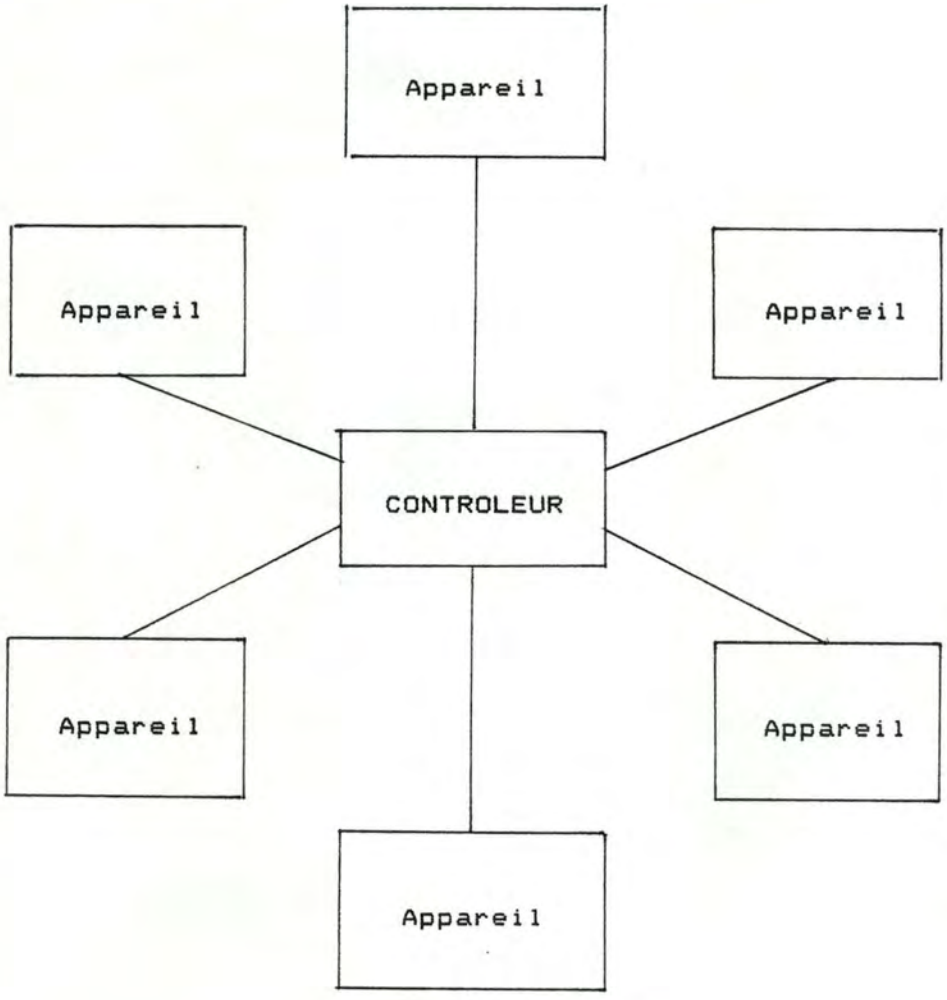


Fig. 1.4.2.0. Connexion d' appareils selon une structure en étoile.

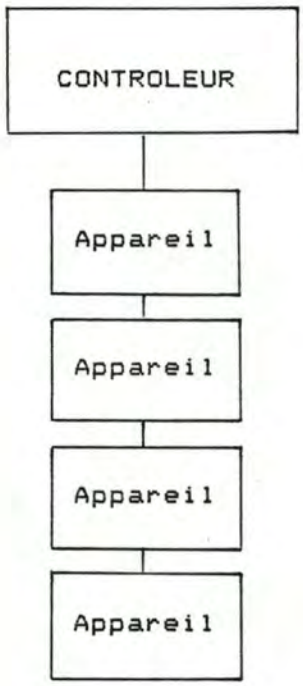


Fig. 1.4.2.1. Connexion d' appareils selon une structure linéaire.

Afin de pouvoir réaliser des transferts de données dans les meilleures conditions possibles, c' est-à-dire avec un minimum d' interférences relatives à l' environnement, une plus grande immunité au bruit et de meilleurs taux de transfert, la norme préconise une limitation des longueurs de connexions entre appareils, qu' elle fixe à vingt mètres au plus, pour l' ensemble de la connexion, et, dans la mesure du possible, à deux mètres entre appareils.

Le connecteur adopté par la norme, est un connecteur de 24 broches, permettant la connection des seizes lignes du bus, ainsi que les lignes de masse électrique et de blindage.

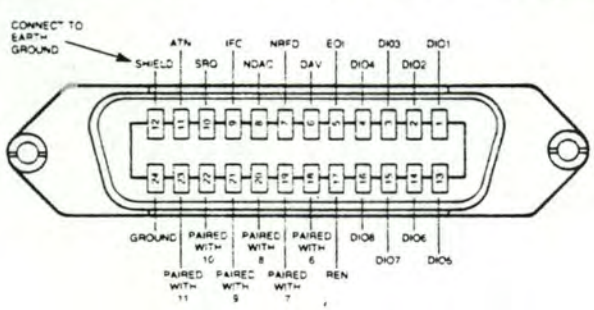


Fig. 1.4.2.2. Connecteur normalisé.

L' assignation des lignes aux broches du connecteur normalisé, se réalise selon le tableau suivant.

broche	ligne	broche	ligne
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EOI	17	REN
6	DAV	18	Masse
7	NRFD	19	Masse
8	NDAC	20	Masse
9	IFC	21	Masse
10	SRQ	22	Masse
11	ATN	23	Masse
12	Blindage	24	Masse log.

Fig. 1.4.2.3. Correspondance des broches du connecteur normalisé aux lignes du bus.

Les lignes de masse sont disposées de manière adjacente aux lignes de contrôle, dans la cas d' une connexion au moyen d' un câble plat traditionnel, ce qui à pour effet d' augmenter l' immunité aux interférences électromagnétiques, ainsi que de diminuer le bruit.

Rappelons que le connecteur adopté par l' I.E.C. en Europe, est un connecteur à 25 broches de type MIL-C-24308, qui est utilisé habituellement à d' autres fins, et qui met alors en oeuvre des tensions et des courants incompatibles avec ceux de cette norme. La majorité des constructeurs européens ont, pour cette raison, décidé d' adopter le connecteur américain, dont l' usage est réservé de facon quasi exclusive à une connexion de type I.E.E.E.-488.

1.4.3. Aspects opérationnels de l' interface.

La norme I.E.E.E.-488, traite des aspects fonctionnels, mécaniques et électroniques de l' interface, dans la mesure ou ces derniers peuvent être établis en faisant abstraction de l' appareil appellé à les supporter.

Une quatrième catégorie d' aspects, concerne la composante opérationnelle de tout interface, nécessaire à la mise en exploitation d' une configuration d' appareils.

Ces spécifications sont particulièrement liées aux systèmes et appareils rencontrés, et se prêtent mal à une normalisation systématique.

Afin d' apporter une réponse à cette situation, l' institution I.E.E.E.-488 publia en 1982, un ensemble de recommandations, relatives aux structures et aux formats à adopter en matière de codification des messages.

Dans la mesure ou aucune convention unique de codification ne pouvait être adoptée, et ce, en raison du caractère trop spécifique de chaque application rencontrée, ces recommandations proposent un ensemble exhaustif de lignes de conduites et d' alternatives, qui aboutissent à la présentation de plusieurs formats de données et de systèmes de codification de l' information, relatifs à l' interface normalisé.

En fait, ces avis reflètent un usage relativement répandu en cette matière, qu' adoptent depuis plusieures années maintenant, de nombreux constructeurs de matériels incorporant l' interface.

Nous reprennons en annexe de ce travail et, à titre d' illustration, des exemples de présentation et de codification de messages de données.

1.5. Le contrôleur I.E.E.E.-488.

1.5.0. Role du contrôleur dans une configuration-système.

Toute configuration d' instruments de mesure fonctionnant selon le système d' interfacage I.E.E.E.-488, nécessite habituellement le concours actif d' un type particulier d' appareil : le contrôleur.

La mission du contrôleur de bus I.E.E.E.-488 consiste à assigner des rôles aux appareils présents dans la configuration, et susceptibles de réaliser les opérations de transfert de données, au moment jugé opportun par le contrôleur.

Les appareils sont impliqués dans un transfert de données ou de communication d' informations d' états, selon une procédure d' adressage menée exclusivement par le contrôleur.

1.5.1. Les fonctions d' interfacage du contrôleur.

La dénomination de contrôleur de bus I.E.E.E.-488 est accordée à tout appareil susceptible d' émettre des commandes d' adressage (dédicacées ou universelles, cfr. supra), à destination d' autres appareils de la configuration, ainsi que de mener une scrutation des appareils sollicitant une attention immediate, selon une procédure prévue par la norme.

Lorsque d' autres appareils présents dans une configuration possèdent cette faculté, ceux-ci doivent alors être placés dans un état d' oisiveté (CIDS : Controller Idle State), afin d' éviter tout problème de contention avec le contrôleur initial, qui prend alors la dénomination de " contrôleur en charge du bus ".

La fonction de contrôle ne sera effectivement supportée, que lorsque l' appareil aura été placé dans l' état SACS (System Control Active State) d' activation.

La fonction de contrôle, une fois activée, sera conservée durant toute la durée des opérations sur le bus, et permettra de réaliser les opérations de passage en mode distant (REN), ou d' initialisation d' appareils (IFC), indépendamment de la faculté d' avoir ou non la charge exclusive du bus.

Le contrôleur est alors appelé contrôleur-systeme (Sytem Controller).

1.5.2. Les états de contrôle.

1.5.2.0. Conventions de représentation des états de contrôle.

La norme I.E.E.E.-488 illustre la fonction d' interfacage d' un contrôleur, au moyen des diagrammes d' états reproduits en page suivante.

Les légendes associées aux différents états, sont reprises dans le tableau qui suit :

C Mnemonics

Messages		Interface States	
pon	= power on	CIDS	= controller idle state
rsc	= request system control	CADS	= controller addressed state
rpp	= request parallel poll	CTRS	= controller transfer state
gts	= go to standby	CACS	= controller active state
tca	= take control asynchronously	CPWS	= controller parallel poll wait state
tcs	= take control synchronously	CPPS	= controller parallel poll state
sic	= send interface clear	CSBS	= controller standby state
sre	= send remote enable	CSHS	= controller standby hold state
IFC	= interface clear	CAWS	= controller active wait state
ATN	= attention	CSWS	= controller synchronous wait state
TCT	= take control	CSRS	= controller service requested state
		CSNS	= controller service not requested state
$\overline{\text{ACDS}}$	= accept data state (AH function)	SNAS	= system control not active state
$\overline{\text{ANRS}}$	= acceptor not ready state (AH function)	SACS	= system control active state
$\overline{\text{SDYS}}$	= source delay state (SH function)	SRIS	= system control remote enable idle state
$\overline{\text{STRS}}$	= source transfer state (SH function)	SRNS	= system control remote enable not active state
$\overline{\text{TADS}}$	= talker addressed state (T function)	SRAS	= system control remote enable active state
		SIIS	= system control interface clear idle state
		SINS	= system control interface clear not active state
		SIAS	= system control interface clear active state

Fig. 1.5.0.0. Mnémoniques des messages et états de contrôle.

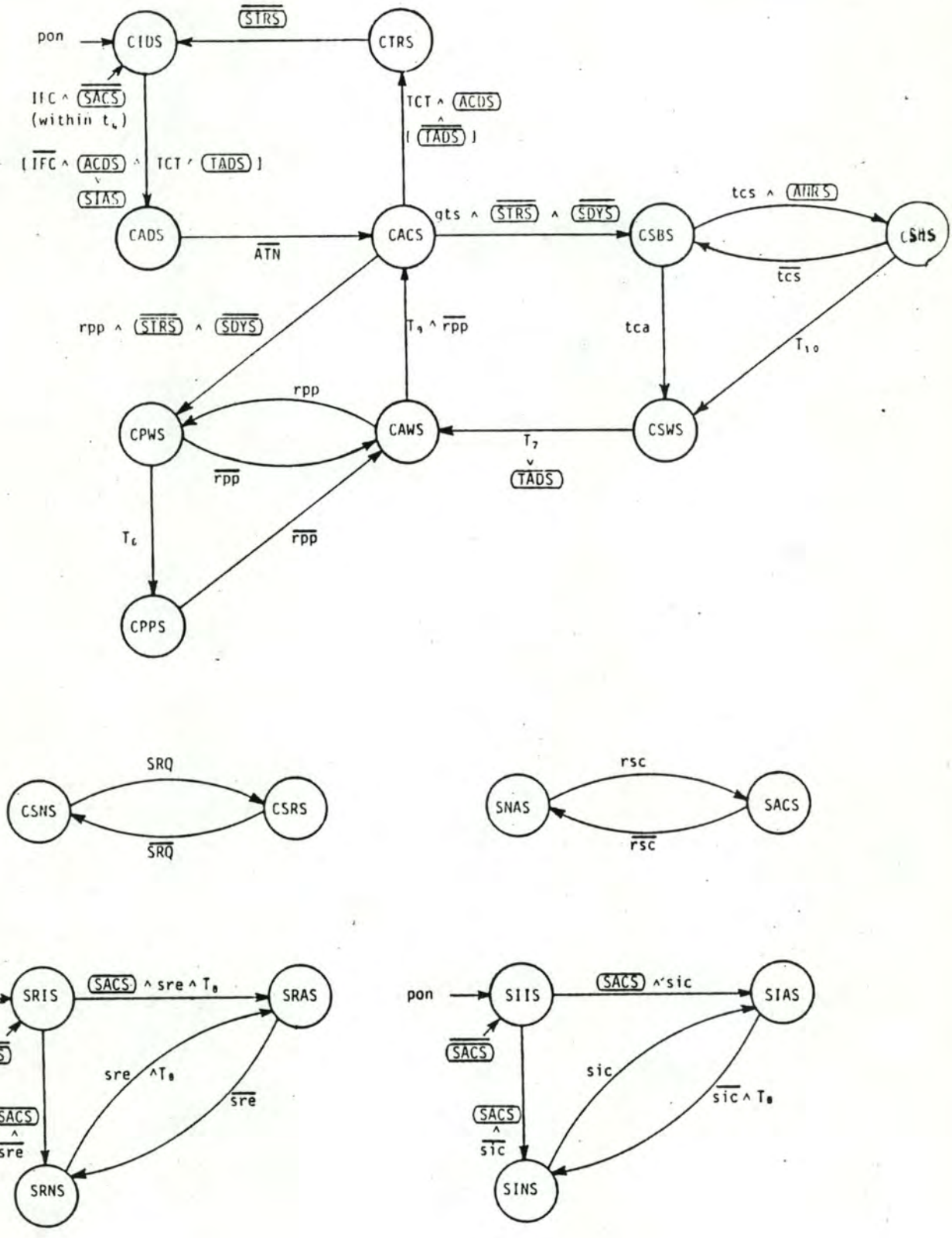


Fig. 1.5.0.1. Diagrammes des états relatifs à la fonction de contrôle.

1.5.2.1. Définition des états de contrôle.

1.5.2.1.0. L' état CIDS (Controller Idle State).

Un contrôleur placé dans l' état d' oisiveté CIDS, abandonne toute possibilité de contrôle du bus, au profit du contrôleur de prise en charge du bus. Une mise sous tension d' un appareil contrôleur a notamment pour effet de placer l' appareil dans cet état.

La transition de l' état CIDS pour l' état CADS (Controller Addressed State) est rendue possible soit :

- au moyen du message TCT (Take Control), issu du contrôleur en charge du bus, pour autant que les états TADS et ACDS aient été préalablement définis, et que le message IFC ne soit pas transmis.

- la présence de l' état SIAS (System Control Interface Clear Active State).

1.5.2.1.1. L' état CADS (Controller Adressed State).

L' état intermédiaire CADS permet au contrôleur de devenir contrôleur en charge du bus, après que le contrôleur courrant ait cessé l' envoi du message ATN sur le bus.

La transition de l' état CADS vers l' état CACS (Controller Active State), est effectuée en portant la ligne ATN à l' état haut (état logique faux).

Une transition de l' état CADS vers l' état CIDS se réalise cependant, lors de la réception du message IFC, et pour autant que le contrôleur ne soit pas le contrôleur système, c' est-à-dire que l' état SACS (System Control Active State) ne se vérifie pas.

1.5.2.1.2. L' état CACS (Controller Active State) .

La fonction de contrôle associée à l' état CACS, permet le transfert de messages issu de la partie fonctionnelle de l' interface (Device Function) sur les lignes de bus.

Ces informations consistent en adresses d' appareils, commandes universelles, ou commandes adressées. L' opération de transfert est effectuée par la fonction SH qui détermine notamment le moment à partir duquel le contenu du message peut être modifié. Durant toute la durée du transfert, le message ATN doit être émis en permanence, ce qui permettra de

transmettre n' importe quel message multiligne que prévoit la norme.

La transition de l' état CACS vers l' état de transfert de contrôle CTRS (Controller Transfer State), interviendra lors de la réception du message TCT dans l' état ACDS et en mode TADS non actif.

La transition de l' état CACS vers l' état d' attente de scrutation en mode parallèle CPWS (Controller Parallel Poll Wait State), s' éffectuera après la réception du message de demande de scrutation en mode parallèle (rpp), pour autant que les états SYDS et STRS aient été activés.

Le retour vers l' état d' oisiveté CIDS ne se réalisera, pour autant que le contrôleur ne soit pas le contrôleur-système (état SACS non vérifié), que lorsque le message IFC sera émis.

Le message gts (go to standby) aura pour effet de placer le contrôleur dans l' état CSBS (Controller StandBy State) pour autant qu' aucun des deux états SDYS et STRS ne soient vérifiés.

1.5.2.1.3. L' état CPWS (Controller Parallel Poll Wait State).

La fonction de contrôle associée à l' état CPWS, consiste à mener une scrutation en mode parallèle par le biais de l' interface, après une période d' attente nécessaire à une stabilisation des ligne de données. Cette opération est réalisée en portant les lignes ATN et EOI à l' état bas (logique "vrai").

La transition de l' état CPWS vers l' état CPPS n' interviendra qu' après l' expiration du délai Tó que prévoit la norme.

Le passage à l' état d' oisiveté CIDS, ne se produira qu' au moment T4, pour autant que le message IFC soit émis, et que l' appareil ne soit pas le contrôleur-système (état SACS non-actif).

Le retour à l' état CAWS se réalise, lorsque le message rpp n' est pas vérifié.

1.5.2.1.4. L' état CPPS (Controller Parallel Poll State).

Dans l' état CPPS, la fonction de contrôle consiste à mener de facon effective, une scrutation des appareils en mode parallèle, et à transmettre les messages PPR issus des lignes du bus, vers la partie fonctionnelle (Device Function)

de l' appareil.

Cette opération se réalise en portant les lignes ATN et EOI à l' état bas (état logique vrai).

La transition de l' état CPPS vers l' état CAWS se réalise, lorsque le message rpp n' est pas verifié.

Le passage à l' état d' oisiveté CIDS, ne se produit qu' au moment T4, pour autant que le message IFC soit émis, et que l' appareil ne soit pas le contrôleur-système (état SACS non actif) .

1.5.2.1.5. L' état CSBS (Controller StandBy State).

La fonction de contrôle associée à l' état CSBS, permet à deux ou plusieurs appareils, de réaliser un transfert de données, sans que le contrôleur n' y prenne une part active.

La ligne ATN doit être portée à l' état haut (état logique faux) par le contrôleur, et le message EOI/IDY ne peut être émis.

La transition de l' état CSBS vers l' état CSHS (Controller Standby Hold State) se réalise à la réception du message tcs (take control synchronously) et dans l' état ANRS.

Le passage dans l' état CSWS (Controller Synchronous Wait State) s' éffectue à la réception du message tca (take control asynchronously).

Le retour à l' état d' oisiveté CIDS, ne se produit qu' au moment T4, pour autant que le message IFC soit émis, et que l' appareil ne soit pas le contrôleur-système (état SACS non actif).

1.5.2.1.6. L' état CSWS (Controller Synchronous Wait State).

La fonction de contrôle associée à l' état CSWS, devrait permettre le passage à l' état CAWS (Controller Active Wait State), après l' expiration de la période T7, afin que le TALKER courant puisse reconnaître le message ATN véhiculé sur le bus. Si la transition vers le nouvel état est issue du message tcs, il appartient à la partie fonctionnelle (Device Function) de l' appareil concerné, de continuer à émettre le message pendant toute la durée associée à l' état. Cette opération aura pour effet de contraindre la fonction AH d' émettre le message RFD, interrompant ainsi le transfert en cours.

Dans cet état, le message ATN doit être transmis avec la signification logique vraie, alors que le message EOI/IDY

doit être transmis (par le contrôleur ou non), avec la signification logique fausse.

La transition vers l' état CAWS, se produit après l' expiration d' une période de temps T7 que prévoit la norme, ou en présence de l' état TADS.

le retour à l' état d' oisiveté CIDS, ne se produit qu' au moment T4, pour autant que le message IFC soit émis, et que l' appareil ne soit pas le contrôleur-système (état SACS non actif).

1.5.2.1.7. L' état CAWS (Controller Active Wait State).

La fonction de contrôle associée à l' état CAWS, permet, au contrôleur, d' entrer dans l' état d' oisiveté CACS, après une période de temps T9, que prévoit la norme.

Ce délai devrait permettre à la ligne EOI de se stabiliser, et d' éviter qu' un appareil n' interprète comme une scrutation en mode parallèle, la signification associée à cet état.

Le message ATN doit être transmis avec la signification logique vraie, et le message IDY avec le signification logique fausse.

La transition vers l' état CACS s' éffectue lorsque le message rpp est transmis avec la signification logique fausse, et après l' expiration de la période de temps T9.

Le passage dans l' état CPWS se produit lorsque le message rpp est transmis avec la signification logique vraie.

Le retour dans l' état d' oisiveté CIDS ne se produit qu' au moment T4, pour autant que le message IFC soit transmis avec la signification logique vraie, et que le contrôleur ne soit pas le contrôleur-système (état SACS non actif).

1.4.2.1.8. L' état CTRS (Controller Transfer State).

Dans l' état CTRS, la fonction de contrôle consiste à envoyer la commande d' adressage TCT à un autre appareil, et à se préparer à passer dans un état d' oisiveté.

Le message ATN doit être alors transmis avec la signification logique vraie, de même que le message TCT qui conserve cette signification, le message IDY est alors transmis avec la signification fausse.

La transition vers 1' état CIDS se réalise, lorsque 1' état STRS ne se vérifie plus, ou au moment T4, à la réception du message IFC, et pour autant que 1' état SACS n' ait pas été activé.

1.5.2.1.9. L' état CSRS (Controller Service Request State).

La fonction de contrôle associée à 1' état CSRS, consiste à signifier à la partie "device function" de 1' appareil, au moyen d' un message local, 1' attention que sollicite un ou plusieurs appareils au sein de la configuration.

Il convient de noter, que la possibilité d' adresser un appareil en mode distant (REMOTE) n' est pas supportée dans ce état.

La transition vers 1' état CSNC (Controller Service Not Requested State) ne se produit que lorsque le message SRQ sera transmis avec la signification logique fausse.

1.5.2.1.10. L' état CSNS (Controller Service Not Requested State).

La fonction de contrôle associée à 1' état CSNS, consiste à signifier à la partie " device function " de 1' appareil au moyen d' un message local, qu' aucun appareil sur le bus ne sollicite 1' attention.

Cet état ne permet pas 1' adressage en mode distant.

La transition vers 1' état CSRS, se réalise lorsque le message SRQ est transmis avec la signification logique vraie.

1.5.2.1.11. L' état SNAS (Sytem Control Not Active State).

La fonction de contrôle associée à 1' état SNAS, consiste à abandonner toute possibilité de contrôle du bus.

Cet état n' autorise pas 1' adressage en mode distant.

La transition vers 1' état SACS se réalise lorsque le message rsc est transmis avec la signification logique vraie.

1.5.2.1.12. L' état SACS (System Control Active State).

La fonction de contrôle associée à l' état SACS, permet d' exercer les opérations de contrôle du bus.

Cet état n' autorise pas l' adressage en mode distant.

La transition vers l' état SNAS se réalise, lorsque le message rse est transmis avec la signification logique fausse.

1.5.2.1.13. L' état SIIS (System Control Interface Clear Idle State).

Cet état est issu d' une mise sous tension de l' appareil contrôleur.

La transition vers l' état SINS s' éffectuera lorsque le message sic sera transmis avec la signification logique fausse, et vers l' état SIAS, lorsque le message sera transmis avec la signification vraie.

1.5.2.1.14. L' état SINS (System Control Interface Clear Not Active State).

Cet état se caractérise par l' impossibilité pour la fonction de contrôle, d' initialiser l' interface.

Le message IFC sera transmis avec la signification logique fausse.

La transition vers l' état SIAS ne se réalisera que pour autant que le message local sic soit transmis avec la signification logique vraie, et la transition vers l' état SIIS ne se produira, que pour autant que l' état SACS ne soit pas actif à ce moment.

1.5.2.1.15. L' état SIAS (System Control Interface Clear Active State).

Cet état se caractérise par l' initialisation de l' interface.

Dès la réception du message IFC, l' ensemble des fonctions d' interfacage du système basculeront dans un état d' initialisation spécifique.

La transition vers l' état SINS s' éffectue dès l' expiration d' une période de temps T8, associée à cet état, et

pour autant que le message local sic soit faux.

Le retour à l' état SIIS, ne se produira que pour autant que l' état SACS ne soit pas actif.

1.5.2.1.16. L' état SRIS (System Control Remote Enable Idle State).

Cet état se caractérise par l' incapacité de la fonction de contrôle, d' assurer un contrôle de la configuration selon le mode distant (REMOTE).

La mise sous tension de l' appareil a alors pour effet d' activer cet état.

La transition vers l' état SRNS ne se produira, que pour autant que le message local sre soit transmis avec la signification logique fausse, et que l' état SACS soit actif.

La transition vers l' état SRAS ne se produira, que pour autant que le message sre soit vrai, l' état SACS soit activé, et que l' état SRIS soit actif endéans une période T8.

1.5.2.1.17. L' état SRNS (System Control Remote Enable Not Active State).

Cet état se caractérise par l' inhibition de la fonction de contrôle en mode distant. Le message REN est alors transmis avec la signification logique fausse.

La transition vers l' état SRAS ne se produira qu' à la réception du message local sre, apres une période de temps T8.

Le retour vers l' état SRIS ne s' éffectue que pour autant que l' état SACS ne soit pas actif.

1.5.1.2.18. L' état SRAS (System Control Remote Enable Active State).

La fonction de contrôle associée à cet état consiste à pouvoir réaliser les opérations du mode distant.

Le message REN sera transmis avec la signification logique vraie durant toute la durée d' activation de l' état.

Le retour vers l' état SRNS se reélisera lorsque le message local sre sera faux, et vers l' état SRIS lorsque l' état SACS ne sera plus actif.

1.5.1.2.19. L' état CSHS (Controller Standby Hold State).

Cet état caractérise l' attente de réception du message DAV avec la signification logique fausse.

La définition de cet état supplémentaire, permet de résoudre le problème de contention pouvant se produire lors de la transimssion simultanée des messages ATN et DAV, lors d' une procédure de prise de contrôle synchrone (tcs).

Durant cette période, le message ATN sera transmis avec la signification fausse, le message IDY avec la signification passive fausse, et le message NUL avec la signification passive vraie.

La transition vers l' état CSWS se produira après l' expiration de la période de temps T10, la transition vers l' état CSBS ne s' éffectuera que pour autant que le message local tcs soit transmis avec la signification fausse, et le retour vers l' état CIDS se réalisera après une période T4, pour autant que le message IFC soit transmis avec la signification vraie et que l' état SACS ne soit pas actif.

1.5.3. Répertoire du sous-ensemble des fonctions de contrôle autorisées par la norme.

Ce répertoire est définis de facon exhaustive par la norme, et comprend l' ensemble des sous-fonctions pouvant être supportées par un appareil contrôleur.

Le concepteur choisira parmi ces fonctions celle qui correspond, selon lui, le mieux au niveau de services qu' il souhaite voir assigner à l' appareil contrôleur.

Le répertoire de ces fonctions est repris en annexe.

1.5.4. Exemples de configurations élémentaires d' appareils.

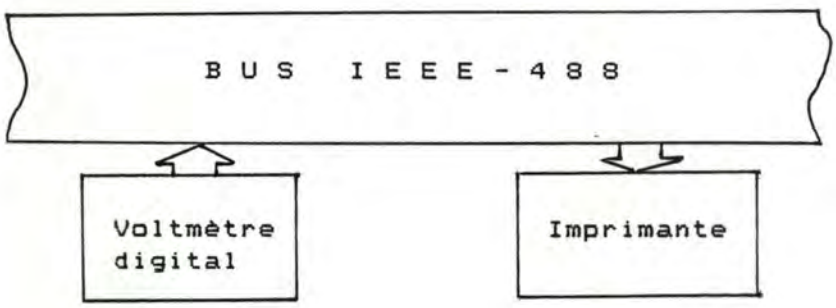
1.5.4.0. Configurations élémentaires sans contrôleur.

Une configuration de base d' appareils supportant la norme I.E.E.E.-488, peut être représentée selon la figure donnée ci-dessous.

Cette configuration comprend deux appareils, un voltmètre digital et une imprimante, qui participent à un transfert de données, sans nécessiter l' intervention d' un contrôleur.

L' échange d' informations se réalise grâce à la fonction de " TALKER " simple, que supporte le voltmètre digital, et qui lui permet de transmettre, de sa propre initiative, des résultats de mesure sur le bus de données, ainsi qu' à la possibilité que possède l' imprimante de se mettre à l' écoute du bus (" LISTENER "), sans pour autant avoir été adressée au préalabe.

Aucun appareil externe ne contrôle l' échange, et la périodicité du transfert est établie par le voltmètre.

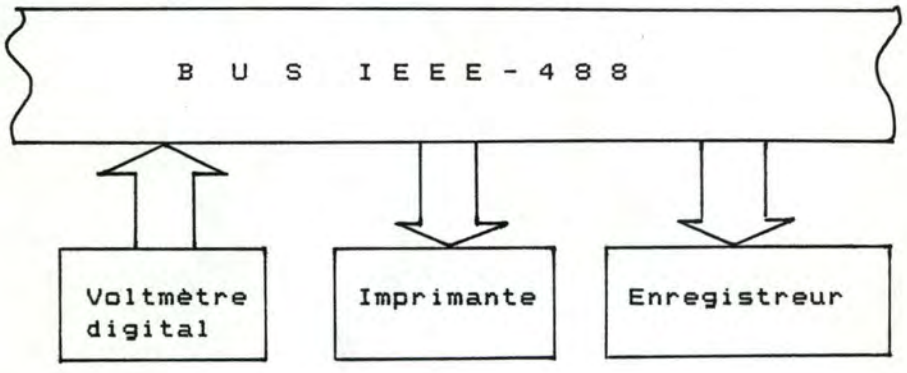


fonction d' interfacage :	Talker	Listener
fonction de l' appareil :	Communication de résultats de mesures.	Edition de données.

Fig. 1.5.4.0. Exemple de configuration élémentaire sans contrôleur (1).

Une extension de cette configuration de base, pourrait consister en l' adjonction d' un appareil destiné à enregistrer les résultats de mesure, aux fins d' archivage.

Ce nouveau type de configuration peut alors se représenter selon le schéma suivant :



fonction d' interfacage :	Talker	Listener	Listener
fonction de l' appareil :	Communication de résultats de mesure	Edition de données	Enregistrement de données.

Fig. 1.5.4.1. Exemple d' architecture sans contrôleur (2).

Cette nouvelle configuration ne nécessite, ici encore, aucun appareil de contrôle externe.

La périodicité du transfert des résultats est fixée par le voltmètre digital, et la vitesse de transmission est adaptée à l' appareil le plus lent de la configuration par le biais de la séquence de synchronisation.

1.5.4.1. Configurations élémentaires de bus incorporant un contrôleur.

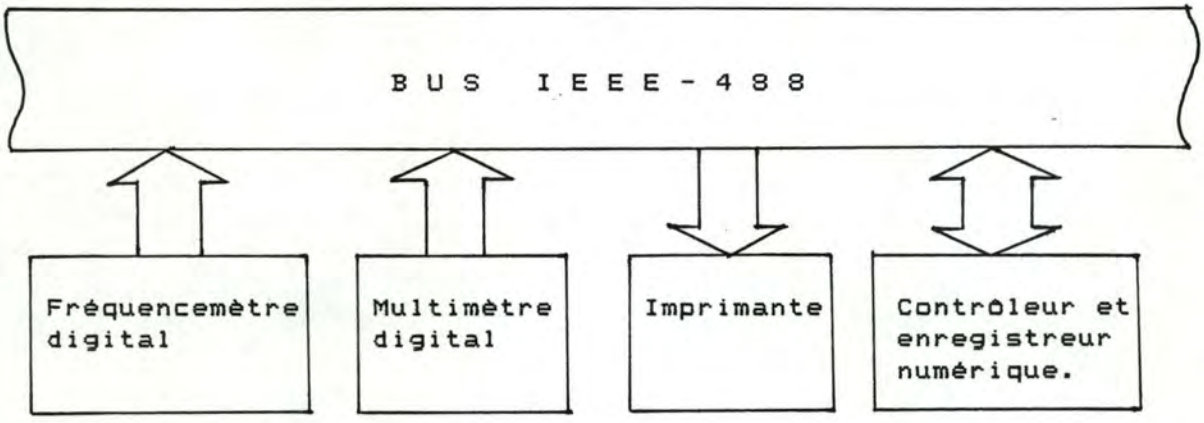
L' introduction d' un contrôleur au sein de la configuration I.E.E.E.-488, a pour objet de réaliser le contrôle des opérations et des flux de données sur le bus, dans les cas ou la complexité de l' application ou le nombre d' appareils interconnectés, nécessitent le recours à un organe supplémentaire, dont la vocation principale est l' agencement des opérations et les transferts de données.

De par le caractère normalisé du bus et des organes qui y sont associés, le choix du contrôleur peut se faire sans se soucier des appareils qui composent la configuration.

Cette particularité va à l' encontre des architectures classiques, ou le contrôleur était étroitement lié aux appareils mis en présence, ou tout au moins dépendant des décisions de conception et de développement de l' architecture rencontrée.

L' introduction de ce nouvel organe au sein de la configuration, offre désormais l' avantage de pouvoir réaliser l' assemblage d' instruments à plusieurs niveaux de compléxité, sans pour autant remettre en cause l' interface de contrôle.

La figure suivante illustre l' introduction d' un contrôleur élémentaire, susceptible d' adresser à tour de rôle deux appareils " TALKER " aux fins de transferts de résultats, à destination de deux appareils " LISTENER ", dont les fonctions sont relatives à l' édition et la mémorisation des résultas aux fins d' archivage.

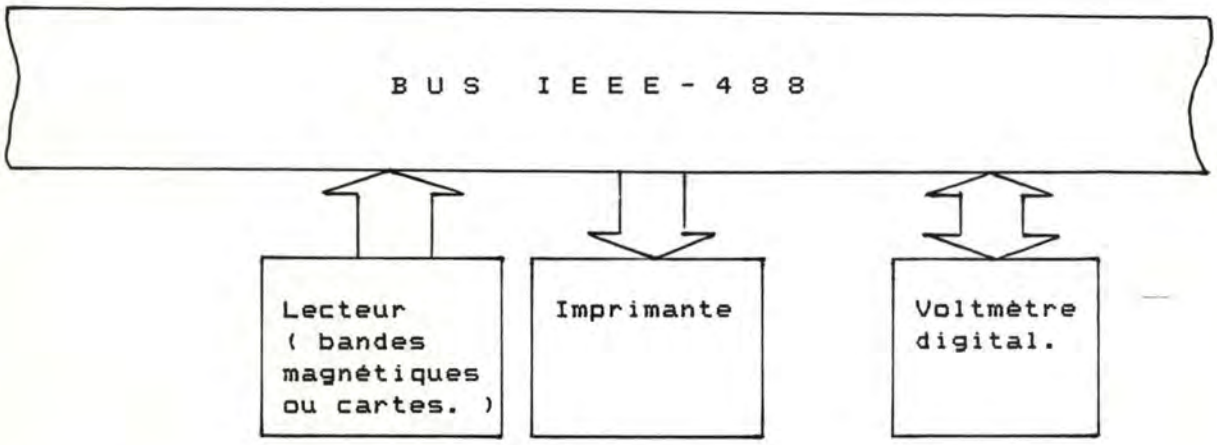


Fonction d' interfacage:			
Talker	Talker	Listener	Controller / Listener
Fonctions de l' appareil:			
Communication de résultats de mesures.	Communication de résultats de mesures.	Edition de données.	Enregistrement de données et contrôle d' appareils.

Fig. 1.5.4.2. Exemple de configuration simple intégrant un contrôleur (1).

L' appareil contrôleur supporte, en plus des fonctions d' assignation de roles, la fonction de " LISTENER ", lui permettant de se mettre à l' écoute du bus de données, afin de consigner les résultats de mesure transmis par les appareils " TALKER ".

Un nouveau type de configuration comportant un contrôleur intégrant la fonction de " TALKER " est représenté ci-dessous:



Fonction d' interfacage :	Talker / Controller	Listener	Talker / Listener
Fonctions de l' appareil :	Emission de paramètres ou commandes. Gestion des transferts.	Edition de données.	Réception de commandes. Emission de résultats de mesures.

Fig. 1.5.4.3. Exemple de configuration élémentaire intégrant un contrôleur (2).

Il s' agit pour ce type de contrôleur de gérer le transfert de résultats s' établissant entre le voltmètre digital et l' imprimante, ainsi que de permettre le contrôle du voltmètre au moyen de commandes de nature opérationelle, réalisant notamment les fonctions de commutation de gammes de mesure et l' activation des mesures au moment jugé opportun par le contrôleur.

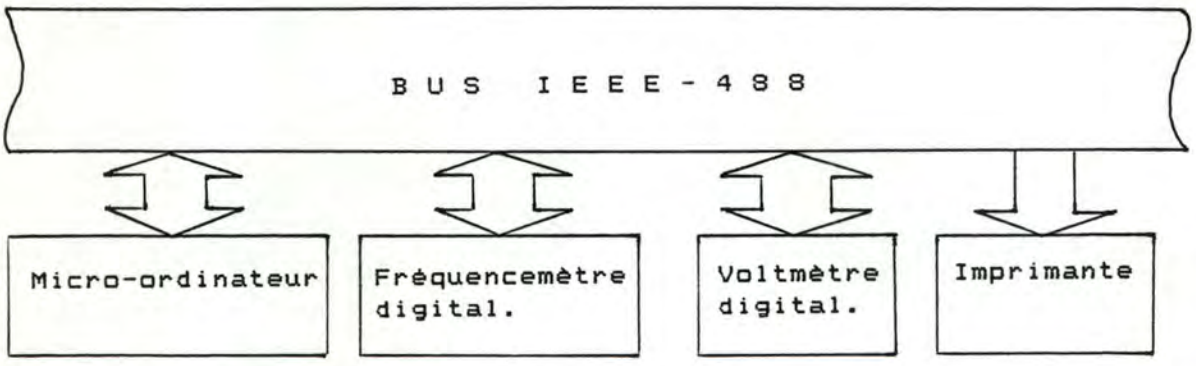
Il convient de noter qu' aucun des schémas présentés jusqu' ici n' incorporent des appareils intelligents de type programmable, tel un micro-ordinateur ou un calculateur.

Cependant, au fur et à mesure que de nouveaux niveaux de complexité apparaissent dans l' élaboration d' une architecture, il convient de prendre en considération les avantages procurés par l' introduction d' appareils intelligents dans la configuration, ceci afin de réaliser plus adéquatement le traitement des données, ainsi que certaines formes de prises de décisions automatisées.

La complexité de l' architecture sera elle même lieé à la nature des applications et opérations que l' on attend de la nouvelle configuration, et non plus dépendante de l' introduction de nouveaux appareils ou d' une décision de reconfiguration du système en place.

Le choix à réaliser entre un calculateur et un ordinateur, ne sera principalement dicté que par des critères de souplesse dans le développement des programmes de contrôle et la possibilité d' intégrer la configuration du système de mesure à l' intérieur d' une configuration de traitement de données.

Le schéma donné ci-après, illustre l' utilisation d' un micro-ordinateur en tant que contrôleur, susceptible de supporter les fonctions de " TALKER " et de " LISTENER ", en plus des fonctions de contrôle du bus et d' assignation de roles des différents appareils.



fonction d' interfacage :			
Controller	Talker / Listener	Talker / Listener	Listener

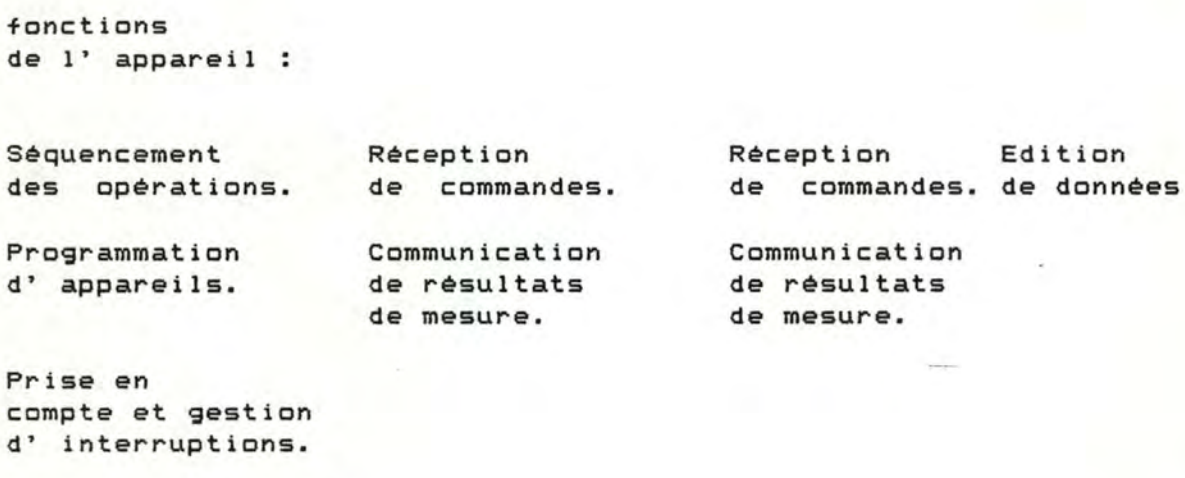


Fig. 1.5.4.4. Exemple de configuration intégrant un contrôleur de type intelligent.

La mission du contrôleur à l' intérieur de cette configuration, pourrait consister en l' adressage et l' activation des appareils de mesure, la programmation des appareils susceptibles de recevoir des commandes (fréquencemètre et voltmètre digital), le séquencement d' opérations de mesure et de communication de résultats, ainsi que la prise en compte de demandes de sollicitation, issues de certains appareils lors de la disponibilité de résultats de mesure, ou encore, lors de la communication de certaines erreurs de programmation ou du contrôle de validité de résultats.

1.5.5. Scénario d' échanges de données.

La figure reproduite ci-dessous reprend un exemple de transfert de données, réalisé sur le bus à l' initiative d' un contrôleur (Fig.1.5.5.0.).

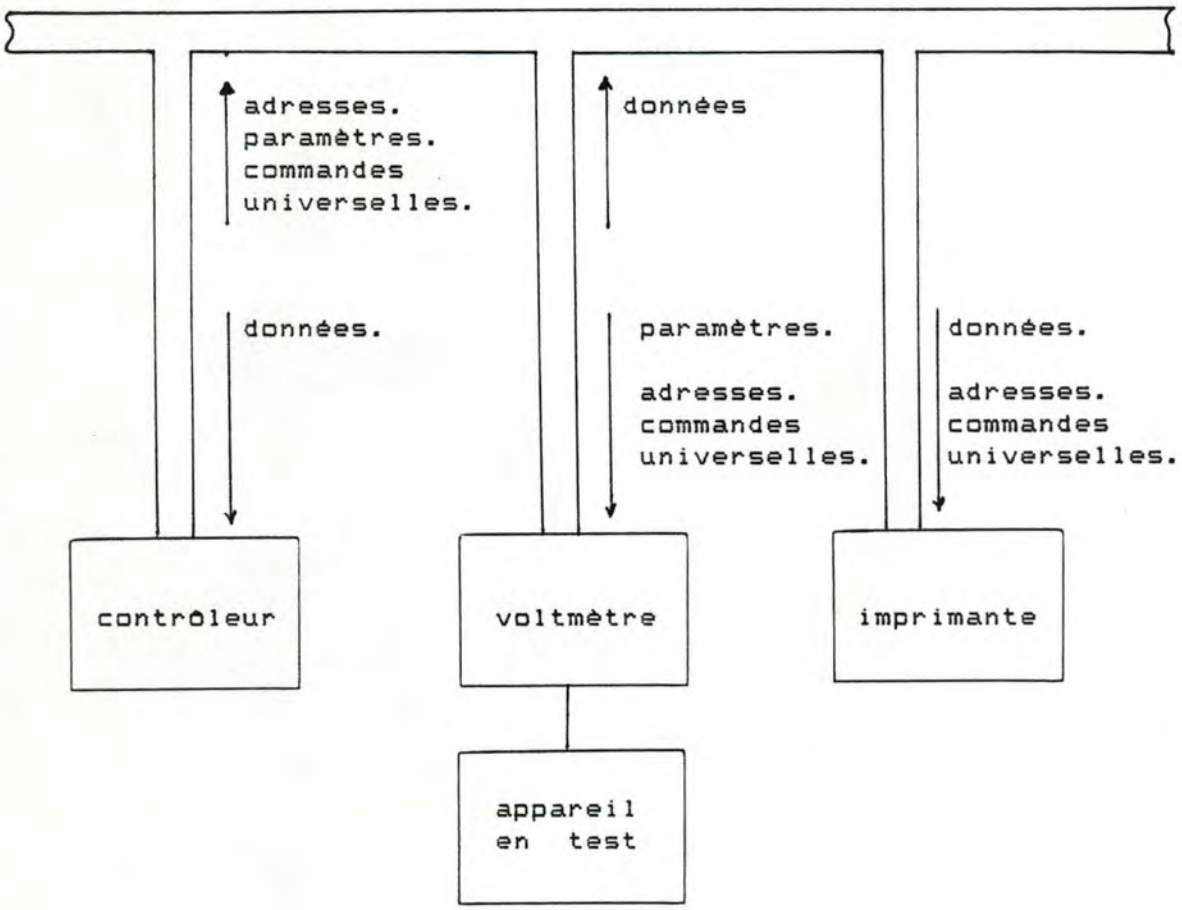


Fig. 1.5.5.0. Echanges de donnéees au sein d' une configuration d' appareils.

Cet exemple illustre l' utilisation d' un contrôleur, ayant pour objectif d' initialiser un processus de mesure, et de transférer les résultats sur imprimante, aux fins d' édition.

La séquence d' échange se réalise conformément aux étapes suivantes :

1. Le contrôleur place le bus d' interfacage dans un état d' initialisation, au moyen du message IFC.

2. Le contrôleur émet le message DCL, afin de placer les appareils dans un état déterminé, lié aux caractéristiques fonctionnelles de l' appareil.

3. Le contrôleur prend la précaution de désadresser tout appareil LISTENER ne devant pas participer au transfet. Cette opération se réalise au moyen du message UNL émis sur le bus de transfert. Il choisit alors l' appareil de mesure participant au transfert, en déposant sur le bus son adresse de LISTENER.

4. Le contrôleur programme l' appareil de mesure, en transférant les commandes relatives au processus de mesure, à destination de l' appareil ainsi adressé.

5. Le contrôleur adresse à nouveau le message UNL, afin d' éviter que toute nouvelle donnée ne soit recue comme commande par l' appareil de mesure, et dédicace ensuite l' imprimante en envoyant son adresse sur le bus et l' invitant, de cette facon, à participer au transfert.

6. Le contrôleur adresse alors l' appareil de mesure en TALKER, afin que ce dernier puisse communiquer ses résultats de mesure au monde extérieur.

7. Une fois les rôles assignés, le contrôleur quitte son rôle de superviseur actif et se met à l' écoute du bus, ceci afin de permettre l' échange de données entre l' appareil de mesure et l' imprimante. Le contrôleur ne reprendra le contrôle qu' à la fin du transfert réalisé par les deux appareils, ou de facon autoritaire, avant la fin du transfert, si une événement inhabituel (déconnexion ou défaillance d' un appareil, par exemple) venait à se présenter.

8. Le contrôleur est prêt à réaliser de nouvelles opérations de controle.

Schématiquement, le transfert se réalise de la facon illustrée à la page suivante (Fig. 1.5.5.1.).

Etat de la ligne ATN		
1	IFC	Emission du message IFC.
1	DCL	Emission du message DCL.
1	UNL	Désadressage des LISTENER actifs.
1	(LAD)	Adressage d' un LISTENER spécifique.
0	(DAB)1	Transfert des paramètres de commande à destination du LISTENER.
	.	
	.	
	.	
0	(DAB)n	
1	UNL	Désadressage du LISTENER actif.
1	(LAD)	Adressage du nouveau LISTENER.
1	(TAD)	Adressage de 1' unique TALKER.
0	(DAB)1	Transfert de données entre le TALKER et le LISTENER.
	.	
	.	
	.	Le transfert se poursuit
	.	jusqu' au moment ou le
	.	TALKER signifie la fin
	.	du transfert au moyen
	.	du message EOI.
	.	Le contrôleur reprend le
	.	contrôle du bus au moyen
	.	de la ligne ATN.
0	.	
	(DAB)n	
1		

fig. 1.5.5.1. Séquence d' échange entre un TALKER et un LISTENER.

CHAPITRE II. Conception et réalisation d' un contrôleur de bus I.E.E.E.-488.

2.0. Introduction.

La première partie de ce travail aborde les concepts d' interfacage tels que développés dans le texte normatif.

Nous reprenons, dans le deuxième volet de cette étude, la démarche utilisée dans les phases de conception et de développement du projet étudié.

Cet itinéraire s' appuie, dans un premier temps, sur l' analyse des avantages relatifs à l' utilisation de micro-processeurs dans le domaine abordé, et plus particulièrement, en liaison avec le bus I.E.E.E.-488.

L' inventaire des outils de développements, matériels et logiciels, permet d' élaborer le profil des outils nécessaires à la réalisation de l' application.

L' analyse des besoins consiste alors à établir, de la meilleure facon possible, les besoins susceptibles d' être satisfaits par l' application future, ainsi que les caractéristiques à prendre en considération au moment de la conception du projet.

Un choix des outils de développement est ensuite mené, compte tenu des contraintes de sélection liées à ce domaine d' applications spécifiques.

L' étude des formes d' implémentation a pour objet de montrer comment, à partir d' une présentation exhaustive des fonctions d' interfacage par la norme, et grâce à une présentation originale donnée sous forme d' automates séquentiels, il est possible d' aboutir à une méthode systématique et simple d' implémentation, que celle-ci soit de nature matérielle ou logicielle.

Cette analyse est complétée par un apercu des possibilités de réduction de certaines fonctions d' interfacage. Ces réductions se présentent sous la forme d' une simplification de graphes de fonctions, et permettent d' aboutir à une implémentation réduite, plus facile à mettre en oeuvre. Cette politique peut s' avérer efficace, principalement en matière de développements réalisés en logique cablée.

La phase finale de réalisation est alors envisagée en tenant compte des caractéristiques et des contraintes de réalisation rencontrées tout au long de la démarche.

Cette étape aboutit à la présentation d' une configuration en matériel destinée à supporter l' application, et de l' architecture logicielle réalisant les fonctions de contrôle proprement dites.

2.1. Considérations de base relatives à la conception de petits systèmes intelligents.

2.1.0. Les avantages liés à l' utilisation des micro-processeurs.

La plupart des contrôleurs intelligents développés actuellement sur le marché, incorporent le plus souvent un ou plusieurs micro-processeurs qui prennent en charge la réalisation de fonctions ne pouvant être supportées jusqu' alors, soit totalement, soit partiellement, sans recours à une logique programmée.

Parmi ces fonctions, citons la possibilité d' auto-calibration ou d' auto-vérification des principaux organes de fonctionnement d' un appareil, que supportent aujourd'hui bon nombre d' appareils de mesure, et qui est rendue possible grâce à un micro-processeur capable de réaliser plusieurs séquences de mesure, et d' établir un diagnostic précis en comparaison à une valeur de référence, que fournit l' organe de mémorisation.

En dehors du fonctionnement interne de l' appareil, le micro-processeur réalise plus adéquatement que les appareils traditionnels, les nombreuses opérations de saisie interactive avec l' utilisateur, établies le plus souvent à partir d' un clavier et d' un écran cathodique.

Par ailleurs, les possibilités de mémorisation et d' édition des résultats constituent un avantage déterminant, qu' offrent aujourd'hui bon nombre de contrôleurs intelligents.

Notons enfin, que le recours au micro-processeur, dans le développement de petits systèmes intelligents, se justifie pleinement, lorsque la complexité de l' application (contrôle de processus industriels complexes, p. ex.) exige une solution plus souple que la solution cablée, onéreuse en matière de conception des circuits et d' acquisition de composants électroniques nécessaires à l' application.

L' utilisation de micro-processeurs dans le développement de petits systèmes, offre l' avantage de la souplesse de développement, et, par conséquent, une réduction des couts inhérents aux développements réalisés en logique cablée.

2.1.1. Généralités relatives au développement de petits systèmes à partir de micro-processeurs.

L' architecture matérielle du petit système que l' on envisage de développer, relève de plusieurs facteurs et, notamment, de l' architecture interne du micro-processeur utilisé.

Dans cette dernière hypothèse, la complexité de l' application orientera, tantot le choix vers une architecture simple, mettant en oeuvre un seul micro-processeur et quelques composants périphériques, tantot vers une structure plus complexe, pouvant nécessiter le recours à plusieurs micro-processeurs, opérants à l' intérieur de relations de contrôle de l' échange plus élaborées.

Toute démarche relative à la conception de petits systèmes intelligents, devrait tenir compte de plusieurs éléments essentiels à l' élaboration du choix défintif :

1. le choix du micro-processeur.

Le choix d' un micro-processeur est déterminant dans la démarche de conception. L' adoption d' un micro-processeur a le plus souvent pour effet de contraindre le concepteur à rechercher lui-même les outils de développement, tant dans le domaine matériel que logiciel, que néglige parfois de lui fournir le fabricant. La disponibilité de ces outils, et l' éventail de leur choix, constitue un élément important, voire dans certains cas décisif, quant à la décision restant à prendre en dernier ressort.

Selon la quantité que l' on envisage de produire, il y aura dans certains cas avantage à orienter le choix du micro-processeur, à l' intérieur d' une catégorie de produits spécifiques. Certains constructeurs proposent à cette fin une gamme de véritables micro-ordinateurs intégrés sur un seul circuit, permettant le développement de larges séries de produits, à partir de deux ou trois circuits seulement.

La complexité de certaines applications amène, dans certains cas, le concepteur à éffectuer la recherche d' un micro-processeur parmi une gamme de produits disposant d' une architecture interne plus élaborée. Cette nouvelle génération de micro-processeurs, offre habituellement l' avantage d' un accroissement sensible dans la vitesse d' exécution des traitements, ainsi qu' une structure d' adressage élargie et un jeu d' instructions spécialement concu aux fins d' être facilement supporté par des langages de haut-niveau.

Cette opportunité offre, dès lors, au programmeur d' application, l' avantage de la rapidité et de la souplesse de programmation, et réduit considérablement, par la même occasion, le temps consacré au développement du logiciel.

L' application de ces quelques critères dans le choix d' un micro-processeur devrait, en principe, réduire l' éventail des produits candidats à la sélection finale.

Afin de pouvoir affiner ce choix définitif, d' autres critères, tels ceux relatifs aux bancs d' essais, au support et à la maintenance assurés par le constructeur, ainsi que certains critères économiques, devraient utilement être pris en compte.

2. le recours aux bancs d' essais.

Les bancs d' essais constituent une étape importante dans le choix d' un micro-processeur.

Ceux-ci consistent habituellemet en un ensemble de programmes ou de sous-routines, destinés à être exécuter comparativement sur des micro-processeurs candidats à la sélection.

Ces test ont pour objet d' évaluer, avec une précision relative, le temps d' exécution nécessité par les programmes mis en oeuvre, et leur taux d' occupation en mémoire. Il convient de signaler l' éfficacité souvent relative de ces tests, issue de la difficulté que l' on rencontre de pouvoir sélectionner le jeux de test pertinent, ainsi que de l' opportunité de pouvoir atteindre le même niveau d' optimisation dans l' écriture des programmes.

De plus, le développement des programmes faisant l' objet d' un test, est habituellement mené à partir d' une connaissance souvent superficielle des micro-processeurs auxquels s' appliquent ces jeux de test.

Un banc d' essai non-concluant peut, dans certains cas, révélé la mauvaise qualité des programmes de test mis en oeuvre, sans, toutefois, permettre d' évaluer objectivement les performances offertes par le micro-processeur. Meme dans les cas ou les tests apparaissent comme négatifs pour un micro-processeur donné, il convient de prendre en considération d' autres critères de sélection, tels la disponibilté de circuits périphériques, la possibilité d' approvisionnement en seconde source que proposent d' autres fabricants, ainsi que la qualité de la documentation accompagnant les composants.

A coté des bancs d' essais réalisés à partir du logiciel, il est utile d' évaluer le matériel en termes de canaux d' entrées-sorties, de possibilités d' interruptions, ainsi que d' autres fonctions, telles l' accès direct à la mémoire, que supportent aujourd' hui la plupart des micro-processeurs disponibles sur le marché.

2.1.2. Micro-processeur et bus I.E.E.E.-488.

A coté des avantages généraux liés à l' utilisation des micro-processeurs, il convient de souligner, à l' étude de la structure du bus I.E.E.E.-488, l' apparente souplesse de mise en oeuvre d' un micro-processeur en liaison avec le bus.

Cette compatibilté semble issue de plusieurs facteurs :

1. La présence d' un bus de données commun.

   Le bus de données de 8 bits du micro-processeur permet le transfert immédiat, en liaison avec le bus I.E.E.E.-488, d' informations codées de type A.S.C.I.I. ou E.B.C.D.I.C., du code BCD, ou de tout autre convention de codification, jugée opportune pour les besoins de l' application.

2. Une structure interne de traitement de 8 bits.

   La structure de l' unité arithmétique et logique sur 8 bits qu' intégrent la majorité des micro-processeurs, facilite la réalisation des opérations logiques, arithmétiques ou de transfert, en liaison avec le bus.

3. Une communication asynchrone.

   Le caractère asynchrone du transfert relatif au bus I.E.E.E-488, permet , quelles que soient les caractéristiques en matière de performance du micro-processeur mis en oeuvre, de résoudre les problèmes d' échanges entre les deux systèmes.

4. La structure des interruptions.

   Les interruptions issues de la gestion du bus I.E.E.E.-488, peuvent être idéalement supportées par le micro-processeur, grâce à sa propre structure de support des interruptions.

5. La gestion des transferts.

   Les lignes de gestion du transfert et de contrôle peuvent être contrôlées au moyen de circuits périphériques spécialisés, opérant sous le contrôle du micro-processeur.

2.2. Inventaire des outils de développement.

Les systèmes de développement constituent les principaux outils permettant de mener à bien la réalisation de logiciels d' application.

En matieré de développement de petits systèmes, les outils de développement sont le plus souvent étroitement liés au type de processeur utilisé, et influencent, de ce fait, fortement les décisions.

Les outils de développement peuvent se subdiviser en matériel et logiciel de développement.

2.2.0. Le matériel de développement.

Il est possible de distinguer deux principales classes d' outils de développement matériel.

La première classe, serait constituée de petits sytemès spécialisés, susceptibles de fonctionner de manière autonome, et disposant d' une unité centrale, d' un terminal, ainsi que de mémoire de masse (disques souples ou durs).

Ces petits systèmes autonomes, dont la vocation principale est le développement de logiciel relatif aux micro-processeurs, offrent le plus souvent la possibilité de supporter plusieurs types distincts de micro-processeurs, ainsi que l' émulation en temps réel.

Ces systèmes ne sont pas liées à l' utilisation exclusive d' un micro-processeur donné, et devraient, pour cette raison, gagner la préférence du concepteur, au moment opportun.

A côté de ces systèmes professionnels, subsistent la classe des micro-ordinateurs à usage domestique, qui offrent, aujourd' hui, un support logiciel et matériel relativement élaboré, permettant de mener à bien le développement d' applications de moyenne importance.

Ces systèmes offrent l' avantage d' un prix d' acquisition peu élevé, ainsi qu' un éventail important de logiciels, mais présentent l' inconvénient de l' absence d' émulation en temps réel, ainsi que la destination exclusive à un type de micro-processeur donné.

En outre, ces systèmes n' offrent qu' une gamme limitée d' appareils périphériques, dont les performances ne peuvent être comparées à celles offertes par les mini ou midi-ordinateurs.

Une deuxième gamme de matériel de développement, comprend l' utilisation d' un ordinateur ou d'un mini-ordinateur disponible sur le site, ou les services offerts par un bureau de service spécialisé.

Le concepteur dispose alors d' un terminal et d' une imprimante, et bénéficie de l' important potentiel, tant matériel que logiciel, qu' offre le système. Les programmes sont alors développés à partir du système, et transférés vers le micro-ordinateur cible, dans la phase de tests.

Les avantages liés à une telle solution, sont étroitement dépendants des performance du système proprement dit, et notamment des temps de réponse associés à l' environement d' une configuration en temps partagé, ainsi qu' à la disponibilité et la qualité des logiciels de développement disponibles.

Une telle solution peut cependant apporter un avantage déterminant, lors du développement d' une gamme de produits-logiciel par une équipe de plusieurs programmeurs, travaillant à la réalisation d' un projet commun, ou nécessitant l' échange de logiciel pour le développement d' applications différentes, faisant cependant appel à des procédures communes.

Ces systèmes proposent à cette fin, une possibilité d' accès à une bibliothèque de logiciel commune à plusieurs utilisateurs, et facilitent le partitionnement du travail de développement en plusieurs équipes.

2.2.1. Le logiciel de développement.

Le choix d' un matériel de développement est étroitement lié au support en logiciel disponible sur le marché.

Cet aspect du choix des outils de développement, s' établit à partir d' une analyse du rapport des coûts de développement du logiciel, sur l' ensemble du cout total du projet, conformément au cycle de vie d' un produit-logiciel.

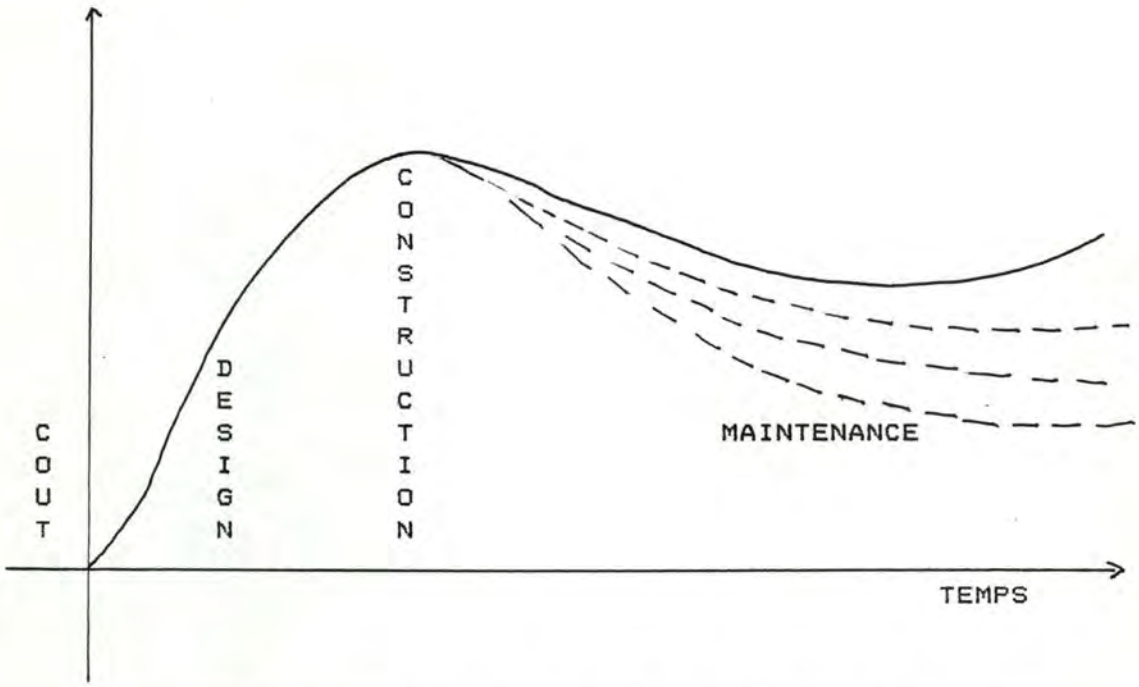


Fig. 2.2.1.0. Cycle de vie d' un produit-logiciel.

L' arsenal des outils relatifs au développement de logiciel à l' usage de petits systèmes, comprend habituellement les éditeurs de texte, les macro-assembleurs et cross-macro-assembleurs, les compilateurs et cross-compilateurs, ainsi que les éditeurs de liens et les chargeurs en mémoire.

Le choix qu' il convient de faire entre les langages de haut niveau et de bas niveau, est principalement dicté par des critères de performance que nécessite le type d' application à traiter.

Le choix d' un langage de haut niveau , offre l' avantage de la facilité et de la souplesse de programmation, et assure une maintenance du logiciel plus aisée. Ces avantages donnent lieu à une réduction sensible des couts de développement, eu égard aux couts nécessités par une programmation en langage assembleur.

Les langages de haut niveau présentent cependant l' inconvénient d' une diminution de performances lors de l' exécution des tâches, ainsi que la génération d' un code peu optimisé, et qui requiert un emplacement en mémoire plus important.

L' évolution de la technologie tend à atténuer considérablement ces inconvénients, en proposant désormais des circuits de mémoire de plus large capacité (256 Kbits), ainsi que des micro-processeurs plus performants (16-32 bits), concus spécifiquement pour pouvoir supporter plus adéquatement les langages de haut niveau.

2.3. Conception de l' application.

2.3.0. Analyse des besoins et spécifications.

Avant toute ébauche de développement, il convient de définir de manière claire et précise, les besoins formulés par les utilisateurs, et les spécifications relatives au nouveau système à mettre en oeuvre.

2.3.0.0. La nature des besoins.

Cette étape dans le développement d' un produit, devrait permettre de répertorier au mieux les besoins manifestés par l' utilisateur, lors de réunions de contact ou d' interviews; les spécifications formulées explicitement et remises à l' analyste, ou les tendances profondes que peuvent refléter certains crénaux de marchés.

Dans le cas de l' application étudiée, les besoins sont apparus lors de contacts menés auprès de distributeurs d' une gamme de produits spécifiques (produits intégrants l' interface I.E.E.E.-488) et faisant apparaitre le besoin de commercialisation d' un appareil, susceptible de supporter les fonctions de contrôle du bus I.E.E.E.-488, selon une procédure familière à l' utilisateur de petits systèmes.

Ces entretiens ont permis de faire apparaitre les les caractéristiques et les besoins suivants:

- appareil de contrôle de bus I.E.E.E.-488, capable de supporter une majorité de fonctions de contrôle, prévues par la norme.
- appareil de type autonome (ne nécessitant pas une connexion à un ordinateur), susceptible de supporter une configuration maximum d' appareils que définit la norme.
- appareil se situant à l' intérieur d' une gamme de prix abordable, pouvant ainsi faire l' objet d' acquisitions par des institutions d' enseignement ou de formation, notamment.
- appareil de type intelligent et interactif, supportant les fonctions de contrôle par le biais de commandes simples et familières à l' utilisateur ("USER FRIENDLY").

2.3.0.1. Les spécifications relatives au système.

Les spécifications relatives au système à mettre en oeuvre, s' appuyent étroitement sur les besoins et les caractéristiques énoncées.

En tentant d' affiner notre analyse, nous pouvons établir le profil du système à réaliser, de la manière suivante :

- petit système intégrant un micro-processeur.

Ce choix prend en considération les avantages réels qu' il y a à choisir un développement à partir d' un micro-processeur, pour ce type particulier d' application (critères de complexité relative de l' application, temps et couts de développement, souplesse de réalisation, etc...).

- petit système autonome.

Cette caractéristique exclut les développements réalisés à partir de mini-ordinateurs, tels ceux proposés jadis par certains fabricants de ce type de matériel.

Notre choix pourrait s' orienter vers une gamme de micro-ordinateurs, pour autant que ces derniers puissent résoudre, sans grande difficulté, les problèmes liés à l' interfacage du bus I.E.E.E.-488.

- petit système interactif.

Les systèmes à base de micro-processeur répondent bien à cette contrainte, et proposent un matériel (circuits d' E/S spécialiés), et des logiciels (langages de commandes,...) appropriés.

- petit système de prix abordable.

Cette contrainte oriente notre choix vers un développement en matériel plus spécifique à l' application étudiée.

L' acquisition, bien qu' à faible prix, d' un matériel ne répondant pas complétement aux besoins de l' application, pourrait, après les nécessaires adaptations, aboutir à une solution dispendieuse, et constituer, de ce fait, un obstacle dans la diffusion du produit.

Il importe, pour ces raisons, de nous orienter vers une solution matérielle, qui réponde plus spécifiquement aux contraintes de l' application étudiée.

2.3.1. Contraintes de sélection et choix des outils de développement.

2.3.1.0. Les outils matériels.

Un inventaire des types d' outils de développement matériel disponible sur le marché a été établi dans un chapitre précédent.

Nous envisageons ici les contraintes liées à l' acquisition d' un matériel de développement relatif à l' application étudiée, et le choix final qui sera fait.

2.3.1.0.0. Les contraintes de sélection.

L' acquisition d' un matériel de développement fait intervenir des considérations de coûts liées à l' achat de ce matériel, ainsi qu' à sa maintenance.

L'ensemble de ces coûts devra être pris en compte, dans l' évaluation du coût total des projets à développer au moyen de ce matériel.

Il importe pour ces raisons, d' envisager l' acquisiton du matériel en tenant compte de l' ensemble des applications à supporter, ainsi que leur nature.

L' application étudiée s' insère, à ce titre, dans le cadre du développement de projets analogues, qui sont le plus souvent réalisés à partir d' une carte de base, développée autour d' un micro-processeur.

Ces projets tendent, habituellement, de résoudre les problèmes posés lors de l' interfacage d' appareils différant le plus souvent dans leurs connexions, et qui constituent, à ce titre, une même gamme d' applications.

La nécéssité de pouvoir réaliser certains projets en équipe, suppose l' utilisation d' un outil de développement commun, devant permettre un meilleur échange du logiciel à partir de bibliothèques de programmes communes, notamment.

Des contraintes relatives aux performances et à la qualité des outils logiciels, nous amène à nous orienter vers des systèmes de moyenne importance, tels des mini-ordinateurs.
Signalons cependant, qu' un nombre croissant de petits systèmes de développement autonome, offrent aujourd' hui un éventail important de logiciels de développement, ainsi que des performances remarquables.

Ces nouveaux outils bénéficient des tout récents progrès technologiques réalisés en matière de micro-processeurs, de logiciels et de mémoires électroniques et de masse, et offrent l' avantage de la souplesse d' utilisation et d' économie en matière de maintenance. De plus, leur caractère de portabilité leur permettent de réaliser de nombreux tests sur le site d' exploitation du nouveau système, et bénéficier ainsi, d' une réduction des couts associés à cette phase du développement du produit logiciel.

2.3.1.0.1. Le choix.

L' orientation du choix vers un mini-ordinateur supportant un système d' exploitation en temps partagé répond, selon nous, le mieux aux contraintes énoncées.

Nonobstant l' absence d' émulation en temps réel, il appert que cette formule facilite le développement d' applications en équipes, ainsi que l' échange de logiciel entre utilisateurs et applications.

Cette option offre par ailleurs une grande souplesse dans la sélection des logiciels de développement, et se prete particuliérement bien à l' évolution de ces outils.

L' absence d' émulation ne constitue pas réellement un problème en soi. Les problèmes rencontrés dans la période de tests, sont le plus souvent issus de mauvaises spécifications dans la phase de conception, et plus rarement de problèmes liés à l' utilisation de composants. De plus, ces problèmes surgissent habituellement lors de nouveaux développements de cartes, mettant en oeuvre de nouveaux circuits. Une fois ces problèmes résolus, les difficultés s' estompent au cours des développements logiciel ultérieurs.

L' utilisation de logiciels utilitaires de mise au point ("debuggers"), constituent, dans ces cas, une alternative avantageuse à l' émulation.

2.3.1.1. Les outils logiciels.

2.3.1.1.0. Les contraintes de sélection.

Le choix du logiciel destiné au développement de l' application, n' est pas étranger aux décisions en matériel qui devront être prises.

Le choix d' un micro-processeur spécifique, contraint habituellement le programmeur, à fixer son choix parmi deux catégories d' outils de développement en logiciel que proposent le fabricant du micro-processeur ou quelques sociétés spécialisées dans le développement de logiciels spécialisés : les (cross)- compilateurs et les (cross)-assembleurs.

Les critères de choix relatifs à ces classes d' outils ont déja été mentionnés précédemment. Signalons cependant, que certaines contraintes plus spécifiques à l' application rencontrée, doivent être prises en compte à ce niveau de développement.

Un exemple de contrainte imposée au langage de haut niveau par l' application étudiée, consiste en un usage intensif des registres interne du micro-processeur contrôleur, lors des procédures d' initialisation, de commande ou d' échanges en relation avec le micro-processeur principal.

De tels transferts ne sont supportés que par certains langages de haut niveau, et ce, au prix d' une réelle dégradation dans les performances globales du système.

Par ailleurs, dans le souci de pouvoir répondre en temps-réel aux évènements survenants sur le bus de transfert, et, conformément aux prescriptions en matière de délais et de temps de réponse fixées par la norme, seules des procédures optimisées en matière de temps d' exécution et de code généré peuvent être retenues. Une solution mixte (adoption des deux types de langages conjointement), peut toutefois être envisagée dans de tels cas, avec le développement de modules écrits en langage de haut niveau pour la partie de l' application qui n' est pas soumise à ces restrictions.

2.3.1.1.1. Le choix.

La nature de l' application étudiée (traitement en temps réel, très orientée entrées-sorties), ainsi que les contraintes de performances évoquées, ont eu pour effet, de porter notre choix sur une classe unique de logiciel: les macro (cross)- assembleurs.

Il convient toutefois de noter, que cette décision fut prise eu égard aux caractéristiques du micro-processeur utilisé, et en fonction de contraintes du moment.

L' évolution constante de la technologie, apparue notamment avec l' éclosion de nouveaux micro-processeurs (16-32 bits), ainsi que l' intégration de plus grandes capacités de mémorisation à l' intérieur de circuits de taille maintenue constante, aurait déjà pour effet de remettre aujourd' hui cette décision en cause et d' envisager l' utilisation de langages à plus haut niveau de développement.

Les langages tels C, Pascal, Modula-2 et Forth répondent, selon nous, le mieux aux critères d' utilisation et de performances énoncés, et devraient, pour ces raisons, gagner plus facilement notre préférence.

Modula-2 constitue par ailleurs un nouveau langage qui introduit aux concepts de développement de programmes par modules, de coroutines, d' accès banalisés en langage machine, de syntaxe systématique et de traitement en temps réel.

Il offre la possibilité de pouvoir développer, tant des drivers-logiciels d' appareils, que des librairies de logiciel-système et de logiciels d' application, et procure une plus grande souplesse en matière de compilation séparées que d' autres langages du même type.

A ce titre, Modula-2 devrait constituer, au même titre d' ailleurs que le langage C, une alternative avantageuse vis-à-vis de langages tels que Pascal ou autres, dans le développement de logiciel système et en temps réel.

Un banc d' essai mené entre plusieurs langages de développement, a par ailleurs permis de mettre en valeur les bonnes performances en temps d' exécution d' un programme écrit en ce langage.

Signalons enfin, que la toute récente éclosion des compilateurs pour petites machines (8 bits), nous a permis d' acquérir, tout récemment, un cross-compilateur 'C', permettant la génération d' un code cible (8085) relativement performant et optimisé.

L' utilisation conjointe de ce langage de haut niveau avec des procédures écrites langage pseudo-assemblé, a déjà permis le développement de nombreuses applications d' interfacage. Des réductions sensibles en matière de temps de développement ont pu être constatées et ce, dès les premières périodes de programmation.

Un nouveau développement de l' application au moyen de ce langage devrait, selon nous, pouvoir être envisagé lors de nouveaux développements, réalisés à partir d' un micro-processeur 16 bits.

2.3.2. Formes d' implémentation des fonctions d' interfacage.

Un chapitre précédent a montrer les possibilités de représentation des fonctions d' interfacage sous forme d' automates séquentiels, comprenant des entrées (conditions primaires de la fonction), des mémoires (états de la fonction), ainsi que des sorties (actions associées à la fonction).

Cette démarche originale, était issue d' une description complète, faite par la norme, de toutes les fonctions d' interfacage, au moyen de diagrammes d' états.

La conception d' une interface I.E.E.E.-488 pourrait idéalement s' appuyer sur cette démarche théorique, afin d' aboutir, dans un premier temps, à la conception d' une interface comprenant les fonctions minimales devant être supportées par celle-ci, et ce, indépendamment des choix d' implémentation qui devront être faits, et, dans un second temps, à la réalisation de l' interface selon l' une des formes d' implémentation qui s' offre au concepteur, eu égard aux critères de performances et aux contraintes de réalisation, dont il devra tenir compte.

2.3.2.0. La démarche d' implémentation.

La concrétisation d' un module séquentiel, dont les fonctions sont exprimées au moyen d' un graphe, ne présente pas de difficultés particulières d' implémentation, qu' elle soit de nature logicielle ou matérielle.

Après avoir définit de facon précise l' automate séquentiel susceptible de supporter la fonction dévolue, la démarche visant à la concrétisation de ces modules fonctionnels, se déroulera de facon différente en fonction de la forme d' implémentation retenue.

A titre d' illustration, il est donné ci-dessous la représentation d' un automate séquentiel informel, sous forme d' un graphe et traduite en table d' états, pour lequel il sera proposé une implémentation de type matériel et logiciel.

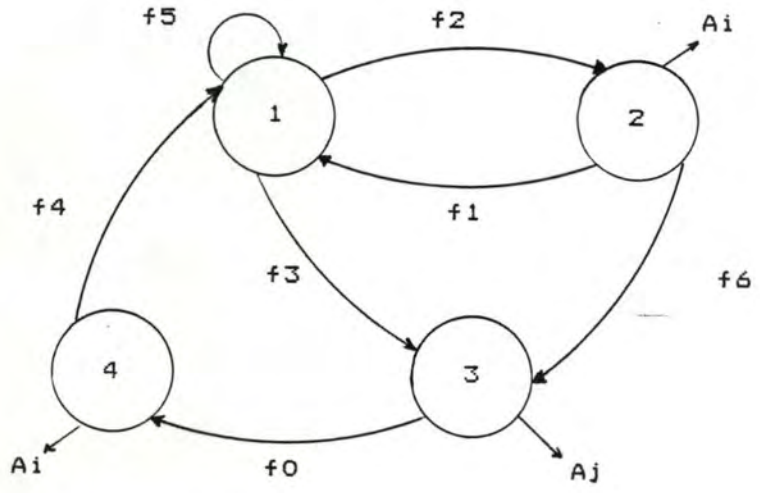


Fig. 2.3.2.0. Diagramme d' états de l' automate.

Ep \ Cp	f0	f1	f2	f3	f4	f5	f6	
1			**2**	3		1		-
2		1					3	Ai
3	4							Aj
4					1			Ai

Fig. 2.3.2.1. Table d' états de l' automate.

Les conventions de représentations adoptées sont :

n : état ou mémoire du systeme
fn : fonction de transition ou condition primaire du systeme.
Al: action associée à un état déterminé.

2.3.2.0.0. Implémentation sous forme matérielle.

Cette démarche part des postulats suivants :

- chaque état du graphe est associé à une mémoire-bascule.

- les transitions entre états sont déterminés par les conditions primaires (entrées de la fonction).

Dans le cas ou une condition primaire se vérifie, la transition d' un état présent (Ep) vers un état suivant (Es) peut alors avoir lieu.

Cette transition se caractérise par :

- l' enclenchement de la bascule relative à l' état suivant (Es).

- le déclenchement de la bascule relative à l' état présent (Ep).

remarque :

Les bascules illustrent les états en tant que mémoires du sytéme. Les circuits logiques représentent les combinaisons logiques associées aux conditions primaires, provoquant la transition par l' enclenchement et le déclenchement d' une bascule.

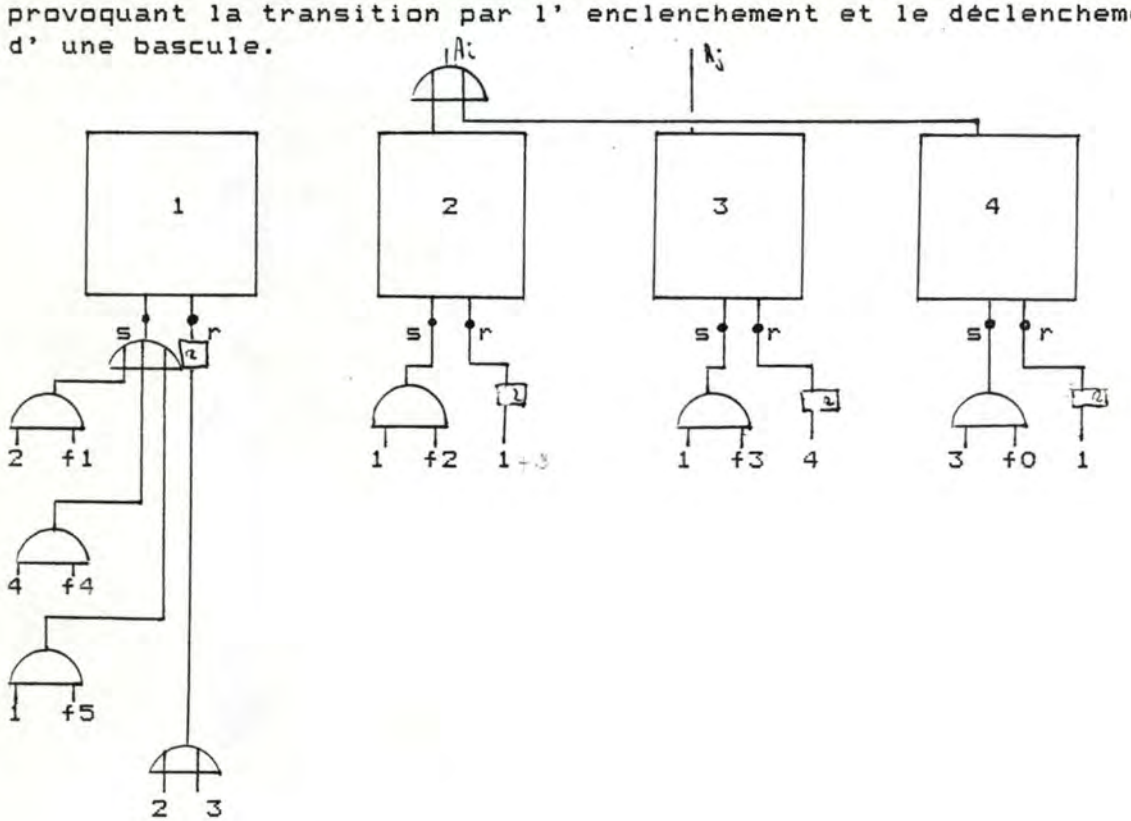


Fig. 2.3.2.2. Implémentation de l' automate sous forme matérielle.

2.3.2.0.1. Implémentation sous forme logicielle.

La représentation du graphe, ainsi que de la table d' état qui y est associée, se fait au moyen de l' ordinogramme suivant :

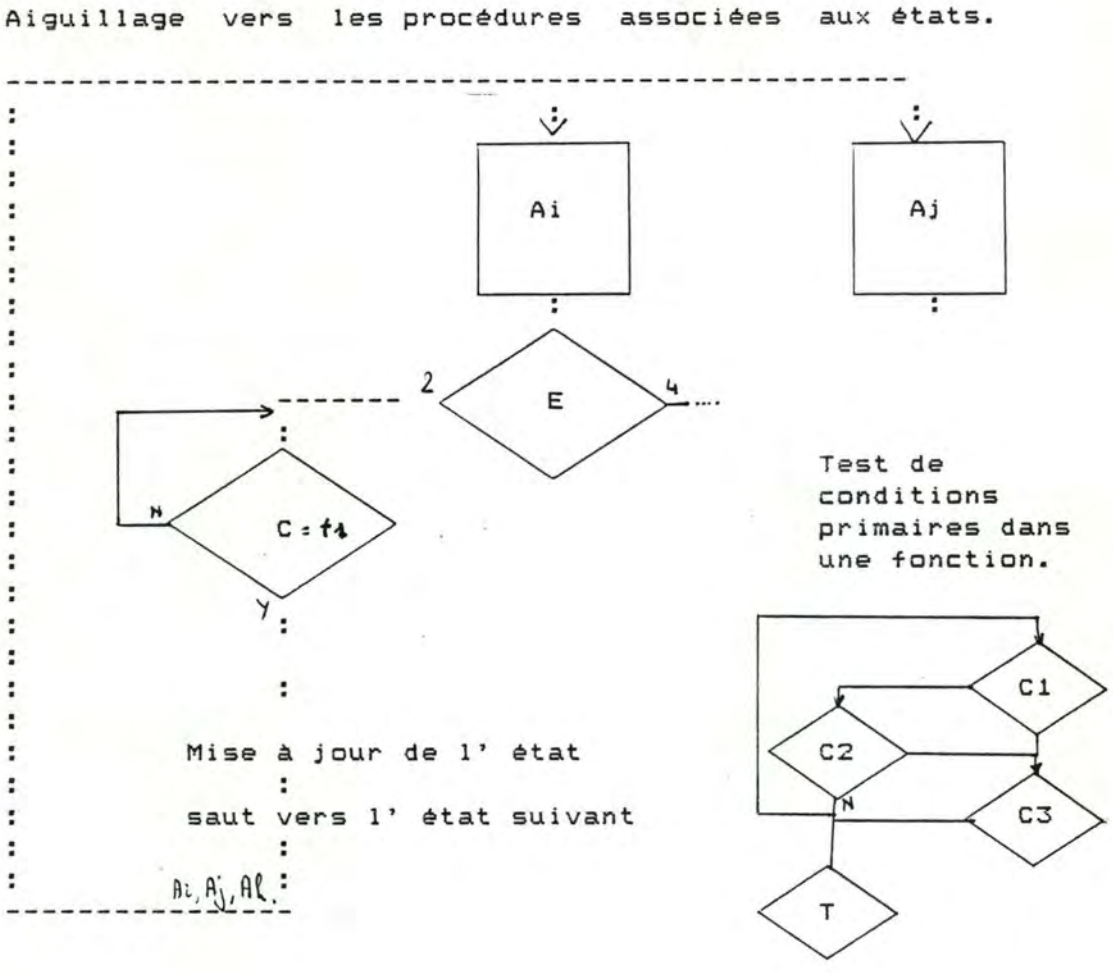


Fig. 2.3.2.3. Implémentation de l' automate sous forme logicielle.

Les conditions primaires s' expriment aux moyens de losanges, marquant la décision à prendre en cas de réalisation ou non de la condition.

Les actions associées aux états prennent, le plus souvent, la forme de procédures ou sous-routines réalisant une action spécifique, et reprises dans l' ordinogramme sous la forme d' un rectangle.

2.3.2.1. Réductions des fonctions d' interfacage.

L' objectif de ce paragraphe, consistera à analyser les graphes normalisés, présentés sous formes d' automates séquentiels, afin de proposer au concepteur d' une interface, des solutions plus simples, et pourtant suffisantes, pour son usage.

Par réduction, il convient principalement d' entendre, une restriction des états associés aux modules séquentiels, aux seuls états considérés comme pertinents pour l' application rencontrée, ainsi qu' une simplification des termes associés aux transitions entre états.

Il sera tenter de démontrer l' utilisation de cette démarche dans la réalisation d' une fonction choisie arbitrairement, parmi l' ensemble exhaustif des fonctions définies par la norme.

Ce choix portera sur la fonction SH (Source Handshakking), qui constitue habituellement une des fonctions de base dans la réalisation d' une interface de type I.E.E.E.-488.

Exemple de réduction : la fonction SH.

Le diagramme d' état relatif à la fonction SH, est repris ci-dessous.

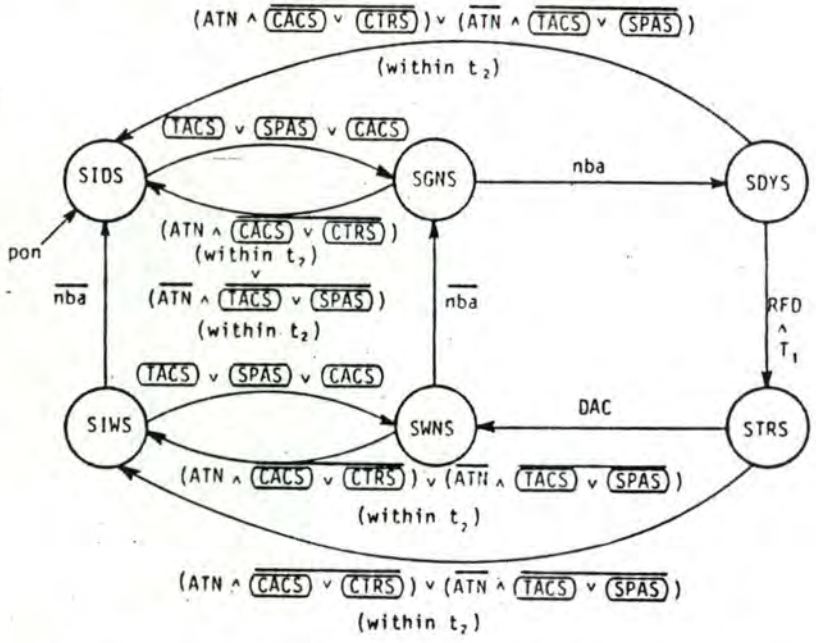


Fig. 2.3.2.4. Graphe d' état normalisé de la fonction SH.

L' étude de ce graphe, fait apparaître les particularités suivantes :

- la présence d' une symétrie horizontale au niveau des couples d' états (SIDS-SIWS) et (SGWS-SWNS) qui les rendent pratiquement équivalent deux à deux.

- une même expression des transitions entre les couples d' états (SIWS-SWNS) et (SIDS-SGNS), et des couples (SIDS-SDYS) et (SIWS-SWNS).

Un examen plus approfondi montre que les états SIWS et SWNS, constituent des états intermédiaires dans la dynamique de la transition, et que seul le message local 'nba' qui marque la disponibilité d' une nouvelle donnée par la partie fonctionnelle de l' appareil, régit les transitions vers ces états d' attente.

Ces états intermédiaires ont, dès lors, pour principal intéret, de permettre la réalisation d' un automate susceptible de réaliser, en tout temps, la synchronisation du transfert à partir du niveau électrique du signal 'nba', issu de la partie fonctionnelle de l' appareil.

Une simplification du graphe pourrait se faire, selon nous, à partir d' un ou plusieurs postulats de base qui imposeraient, par exemple, que la signification logique associée à la ligne 'nba' soit fausse, une fois l' état STRS atteint.

Cette facon de procedéer, offrirait l' avantage de pouvoir supprimer du graphe normalisé, les états d' attente SIWS et SWNS, afin de permettre de présenter un nouveau graphe réduit, plus facile à implémenter.

L' inconvénient de cette pratique, réside toutefois dans une moins large indépendance entre les parties fonctionnelles et d' interfacage de l' appareil et va, à ce titre, à l' encontre de l' esprit d' indépendance entre les composantes d' un appareil, que prône la norme.

Nous reprenons, ci-dessous, le graphe réduit de la fonction, après avoir opéré cette simplification.

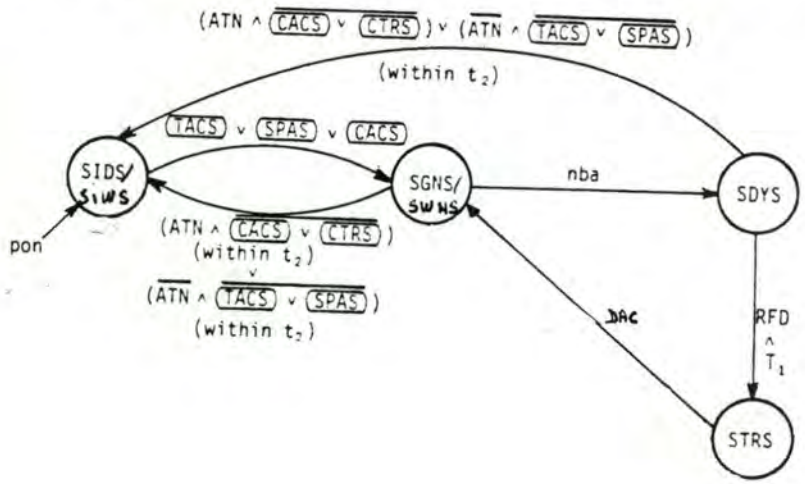


Fig. 2.3.2.5. Exemple de graphe d' état réduit de la fonction SH.

Remarquons que cette simplification a été choisie ici arbitrairement, en imposant certaines contraintes de réalisation, que le concepteur pourrait se refuser à appliquer.

D' autres formes de simplification sont toutefois possible, et peuvent faire l' objet d' autres méthodes de réduction, que celle adoptée plus haut.

Rappelons que cet exemple n' a ici pour objet que de montrer comment, à partir de graphes normalisés, le concepteur peut arriver à des formes réduites du graphe, et aborder la phase d' implémentation avec des solutions plus faciles à mettre en oeuvre.

2.4. Réalisation de l' application.

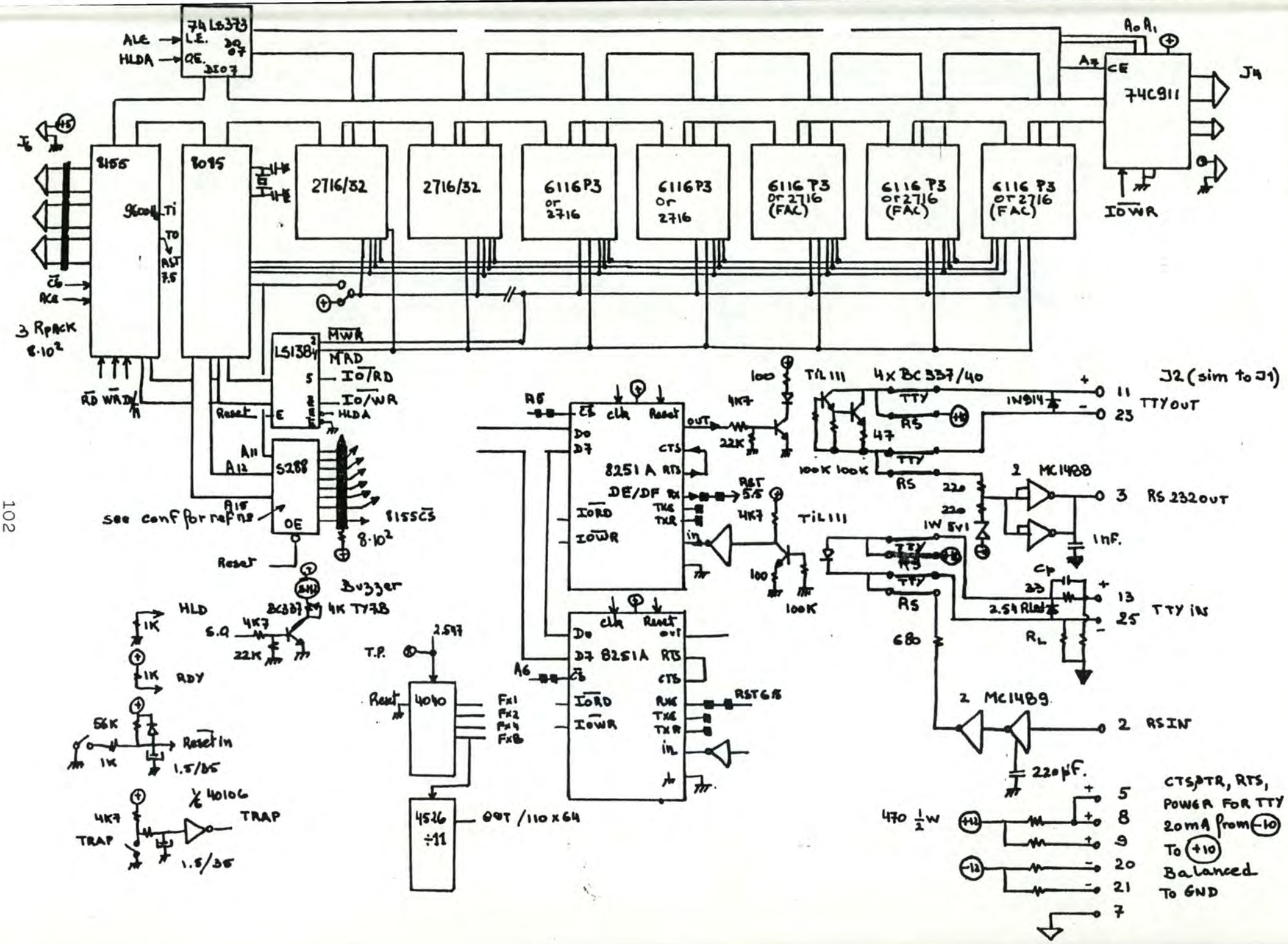
Le chapitre précédent a permis de déterminer le profil de l' application que nous envisageons ici de mettre en oeuvre, ainsi que les contraintes et les choix relatifs aux moyens.

Nous proposerons ici un matériel destiné à supporter l' application de contrôleur I.E.E.E.-488, et tenterons d' élaborer l' architecture logicielle correspondante.

2.4.0. Choix du matériel destiné à supporter l' application.

2.4.0.0. Ebauche d' architecture matérielle.

Compte tenu des avantages relatifs à l' utilisation du micro-processeur dans le domaine qui nous concerne, et en tenant compte de la complexité de l' application à développer, nous reprennons ici le schéma d' une carte électronique comprennant un micro-processeur, que nous destinons habituellement aux développements d' applications similaires.


74LS373
ALE
HLDA
A0 A1
A2
CE
74C911
J4
8155
8085
2716/32
2716/32
6116P3 or 2716
6116P3 or 2716
6116P3 or 2716 (FAC)
6116P3 or 2716 (FAC)
6116P3 or 2716 (FAC)
IOWR
3 RPACK 8.10²
LS138
MWR
MRD
IO/RD
IO/WR
HLDA
RD WR
Reset
A11
A12
A13
5288
OE
see conf for refns
8155CS
8.10²
Reset
Buzzer
BC337
4K TY78
HLD
1K
1K
RDY
5.9
4K7
22K
56K
1K
Resetin
1.5/65
½ 40106
4K7
TRAP
TRAP
1.5/35
T.P.
2.5V
Reset
4040
Fx1
Fx2
Fx4
Fx8
4526 ÷11
OUT /110 x 64
A6
8251A
Do
D7
CTS
RTS
IORD
IOWR
RXR
TXR
in
RST6.5
8251A
DE/DF
RST 5.5
4K7
100
TIL111
100K
100
4K7
22K
TIL111
4X BC337/40
TTY
RS
47
100K 100K
1N914
J2 (sim to J1)
11
TTYOUT
23
2 MC1488
220
220
1W 5V1
3
RS232OUT
1nF.
Cp
33
2.5K RLadj
13
TTY IN
25
RL
680
2 MC1489
2
RSIN
220µF.
470 ½W
+12
-12
5
8
9
20
21
7
CTS,DTR, RTS, POWER FOR TTY
20mA from -10
To +10
Balanced
To GND

Il s' agit d' une carte unique, integrant un micro-processeur (8085 - INTEL), ainsi que ses composants périphériques, qui réalisent les fonctions de support des entrées-sorties séries (8251) et parallèles (8251), ainsi que la temporisation.

La mémoire se compose de circuits de mémoire vive de type statique et d' une capacité de 4k octets, extensible à 10k octets sur la carte même, ainsi que de circuits de mémoire morte, contenant les programmes d' applications.

Cette configuration de base permet de supporter aisément la majorité des applications d' adaptation et d' interfacage rencontrées.

Lorsque certaines applications nécessitent, de par leur spécifications, une extension des fonctions supportées par la carte de base, on procède alors, soit à une extension de la version de base, au moyens de modules ou de cartes d' extensions, notamment, soit au développement d' un prototype original.

- extension de la version de base :

L' extension de la version de base se réalise habituellement, par l' adjonction de modules au circuit existant.

Ces modules intégrent les fonctions n' étant pas supportées par la version de base, et qui répondent de ce fait au souhait d' un client particulier (CUSTOMIZED MODULES).

Il en est ainsi des modules d' adjonction de mémoire non volatile, que nécessitent certaines applications, ou d' adaptations particulières à un certain type de matériel.

- développement d' un prototype.

Ce procédé constitue une étape importante dans le développement d' une interface. Il nécessite de concevoir, soit totalement, soit partiellement, une nouvelle carte en réalisant l' introduction ou l' adjonction de nouveaux composants, afin de répondre aux spécifications de l' application rencontrée.

Cette phase se caractérise par une importante augmentation des coûts de développement sur le plan matériel, mais aussi sur le plan logiciel (développement de nouvelles procédures de programmes dans la gestion des nouveaux circuits, tests des circuits,...).

Le prototype n' atteint le stade de la production finale, qu' aprés avoir été soumis aux nécessaires tests d' acceptation et de performances propres à l' application.

Compte tenu des contraintes énoncées, le développement d' un prototype ne se sera décidé, qu' après avoir envisagé toutes les possibilités qu' offre la carte de base.

Les objectifs sous-jacents aux développements réalisés à partir de la carte de base, peuvent s' énoncer de la manière suivante :

- réduction à un niveau acceptable, des coûts de développement nécessités par une application.

- simplification des développements en logiciels qui, de par leur nature (application en temps réel, le plus souvent), sont étroitement liés à l' implémentation des circuits électroniques mis en oeuvre.

- faciliter la rédaction et la mise à jour de la documentation, en tentant de limiter le nombre de réalisations différentes.

- minimiser les coûts de développement en matériel que nécessitent la conception et la réalisation de nouvelles cartes, destinées, habituellement, à de petites séries de production.

2.4.0.1. Contraintes de réalisation.

La réalisation de l' interface contrôleur, prend en considération plusieures contraintes de réalisation, que nous rencontrons habituellement lors de l' étude de projets similaires.

2.4.0.1.0. Aspects de la compatibilité avec le matériel existant.

L' adoption de la carte électronique de base, présentée dans le paragraphe précédent, n' a put etre retenue pour les raisons suivantes:

- les dispositifs d' émission associés au bus I.E.E.E.-488, doivent comporter des circuits conducteurs de ligne ("DRIVERS"), susceptibles de supporter la charge de 48 mA que fixe la norme.

Les circuits périphériques de transmission parallèle (8155) que possède la carte de base, ne répondent pas à cette exigence, et n' offrent aucune protection particulière vis-à-vis du bus de transfert.

- la possibilité de réaliser un transfert en mode parallèle, selon la procédure classique de synchronisation que permet le circuit périphérique de la carte de base, ne peut s' appliquer au mode de transfert asynchrone à trois fils que prévoit la norme. Les transferts ne peuvent se réaliser, qu' en utilisant le circuit en mode dégradé, c' est-à-dire, en procédant selon une procédure de scrutation des lignes de gestion du bus, au détriment des réels avantages, qu' offre le circuit, en matière de gestion du transfert.

- La diversité des tâches devant être supportées par le micro-processeur, ne permet pas de satisfaire aux exigences relatives au temps de réponse et au taux de transfert que prévoit la norme. De la même manière, le seul mode de transfert autorisé dans un souci de performance (la scrutation), se prête mal à une gestion de tâches concurrentes, que pourrait utilement supporter l' interface.

2.4.0.1.1. Aspects économiques.

Ce problème est habituellement rencontré lors de la production en petite série, d' une nouvelle version d' interface, destinée à satisfaire un problème particulier.

La réalisation d' une nouvelle carte suscite un accroissement de frais fixes par nature, qu' il convient d' incorporer dans le coût de chaque unité du produit à développer.

Dans le cas d' un produit banalisé, l' investissement consenti sera réparti sur un nombre plus importants d' exemplaires du produit, eu égard à l' espérance de vente attribuée.

Le développement d' un contrôleur I.E.E.E.-488 répond à ce souhait de développement d' un produit banalisé.

2.4.0.1.2. Aspects de la performance.

Ce volet concerne les performances globales du système à mettre en oeuvre.

Le développement tout récent de circuits spécialisés dans la gestion du transfert parallèle I.E.E.E.-488, apporte une solution au problème de performance qui se posait jusqu' alors, aux contrôleurs réalisant les fonctions de gestion de bus, selon une procédure de scrutation de lignes.

Les nouveaux circuits spécialisés, sont susceptibles de fonctionner de manière autonome (c' est-à-dire sans faire appel à un micro-processeur pour leur gestion propre), et intégrent la plupart des fonctions que prévoit la norme.

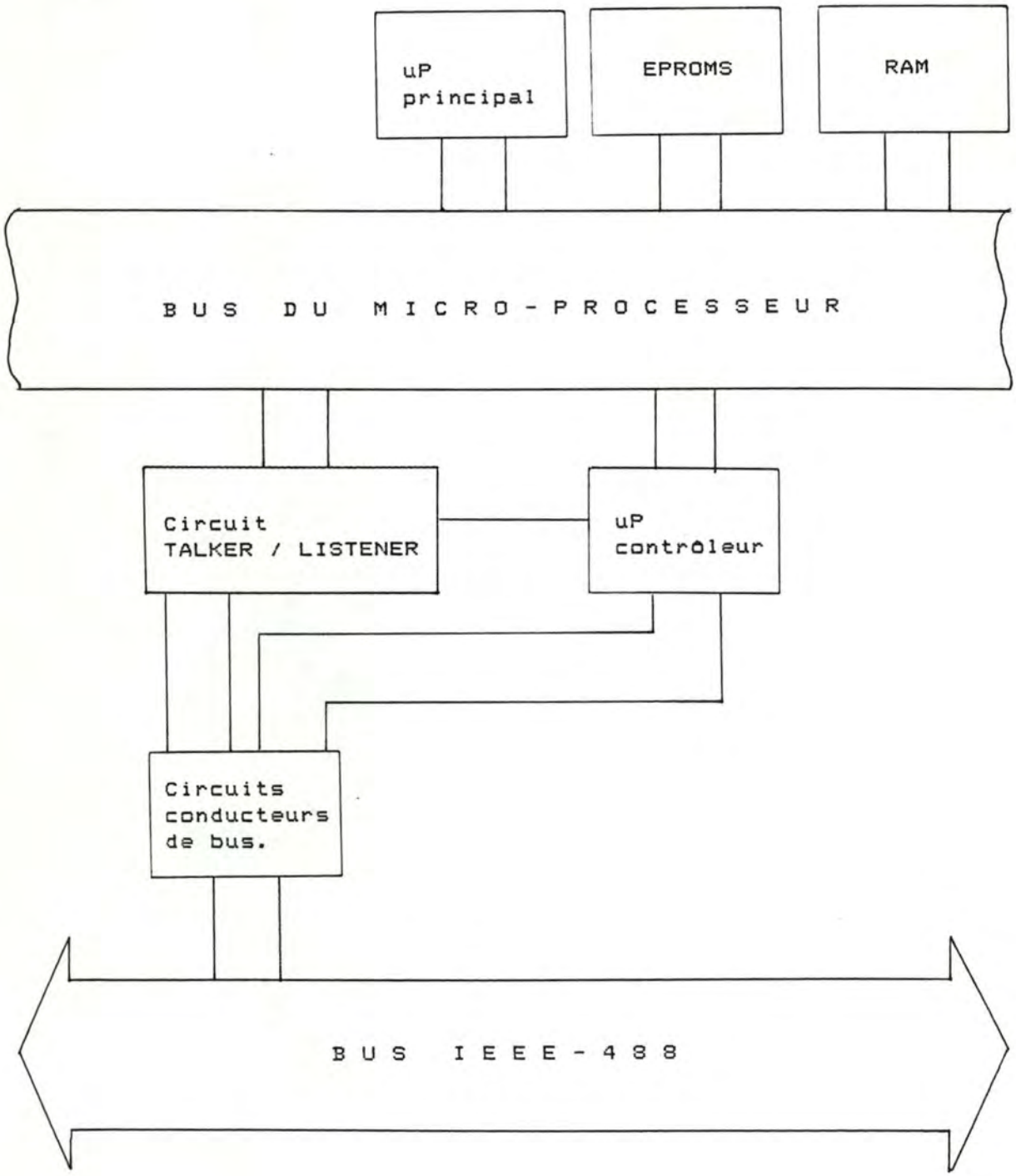
D' autres circuits, réalisent les fonctions de contrôleur, à partir de versions spéciales de micro-processeurs pré-programmés, permettant ainsi de réaliser les fonctions de contrôle et de gestion du bus, parallèlement à l' exécution d' autres tâches, que réaliserait le micro-processeur principal.

Ces circuits effectuent de facon asynchrone, la gestion des transferts, ce qui permet l' exécution d' autres tâches, moins spécialisées, en quasi-simultanéité.

2.4.0.2. Choix final.

Compte tenu des contraintes de réalisations rencontrées, le développement d' une nouvelle carte à été envisagé.

Nous reprennons ici son architecture générale.



L' architecture proposée intégre, à l' architecture de base, de nouveaux circuits :

- le contrôleur: il s' agit d' un second micro-processeur qui réalise les opérations de contrôle définies par la norme, dans une relation "esclave" vis-a-vis du micro-processeur principal.

  Ce circuit est constitué d'un micro-processeur specialisé, disposant de mémoire et de lignes d' entrées-sorties, afin de pouvoir réaliser de manière autonome la diversité des tâches que prévoit la norme.

  Son fonctionnement asynchrone dans sa relation avec le micro-processeur principal, permet l' exécution des taches courantes en parallèle et en simultaneité.

  Le contrôleur est utilisé conjointement avec un ciruit "TALKER-LISTENER" afin de pouvoir supporter l' éventail complet des fonctions I.E.E.E.-488.

- le circuit "TALKER-LISTENER":

  Ce circuit réalise la plupart des opérations ne relevant pas des fonctions de contrôle.
  Ses principales caractéristiques comprennent:

  - le support des fonctions "SOURCE AND ACCEPTOR HANDSHAKE".

  - la réalisation des fonctions de "TALKER" et "LISTENER" avec la possibilité de définir le mode d' adressage étendu.

  - le support des fonctions "SERVICE REQUEST", "PARALLEL POLL", "DEVICE CLEAR", "DEVICE TRIGGER", "REMOTE", "LOCAL".

  - un taux de transfert programmable.

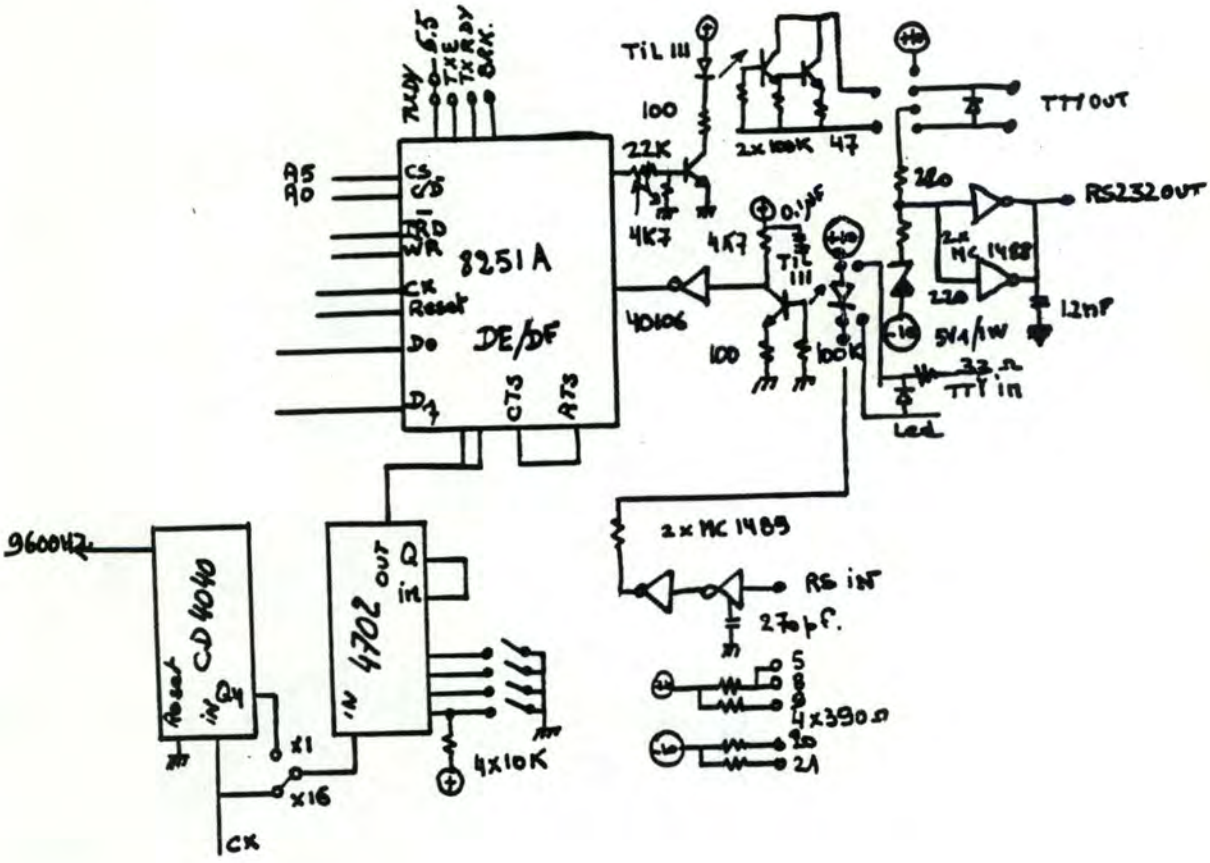
- les circuits conducteurs de bus:

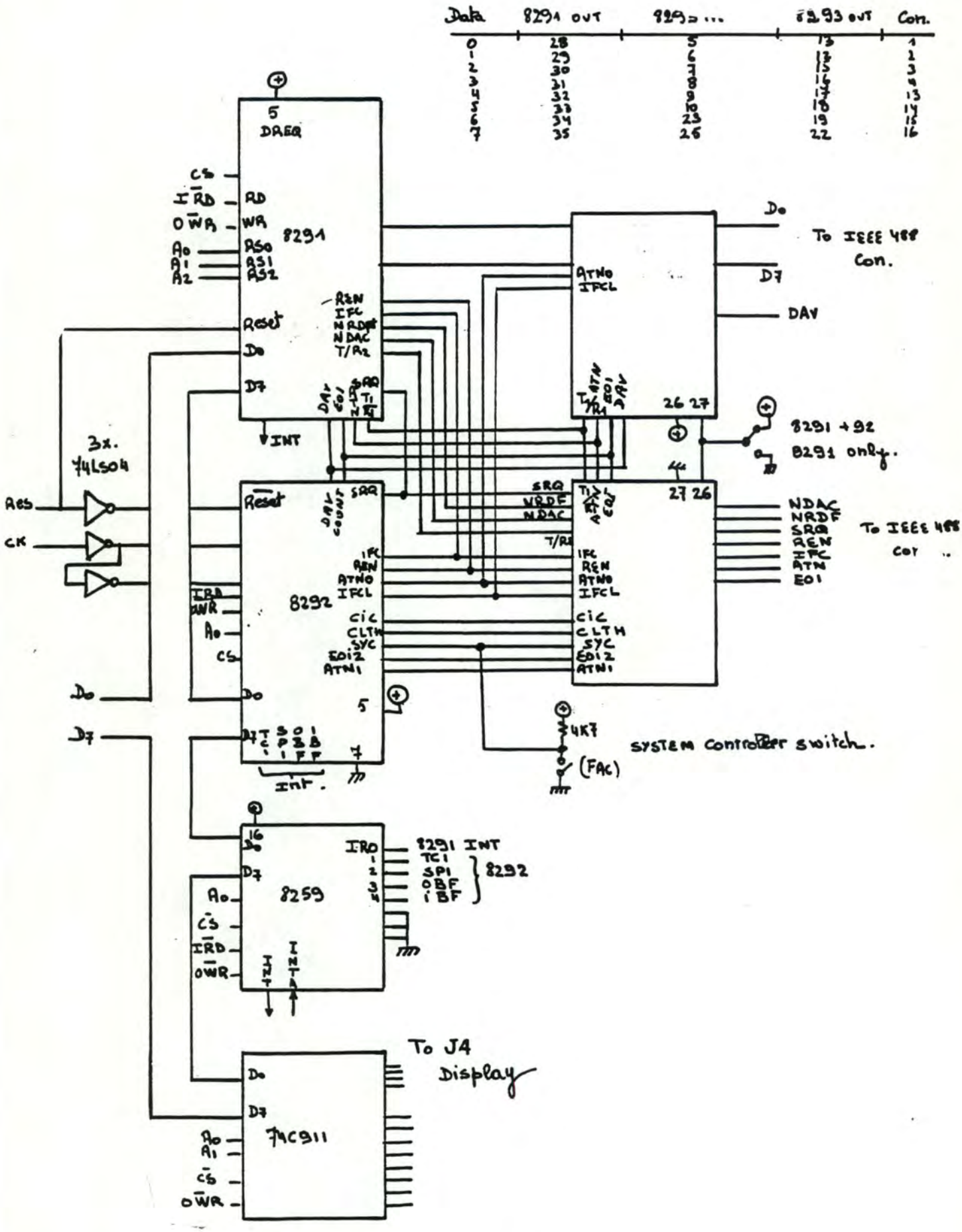
  Ces circuits implémentent les fonctions de contrôle en tension et en courant des lignes de bus selon les specifications I.E.E.E.-488, et intégrent une logique supplémentaire destinée au fonctionnement des circuits contrôleur et "TALKER-LISTENER".

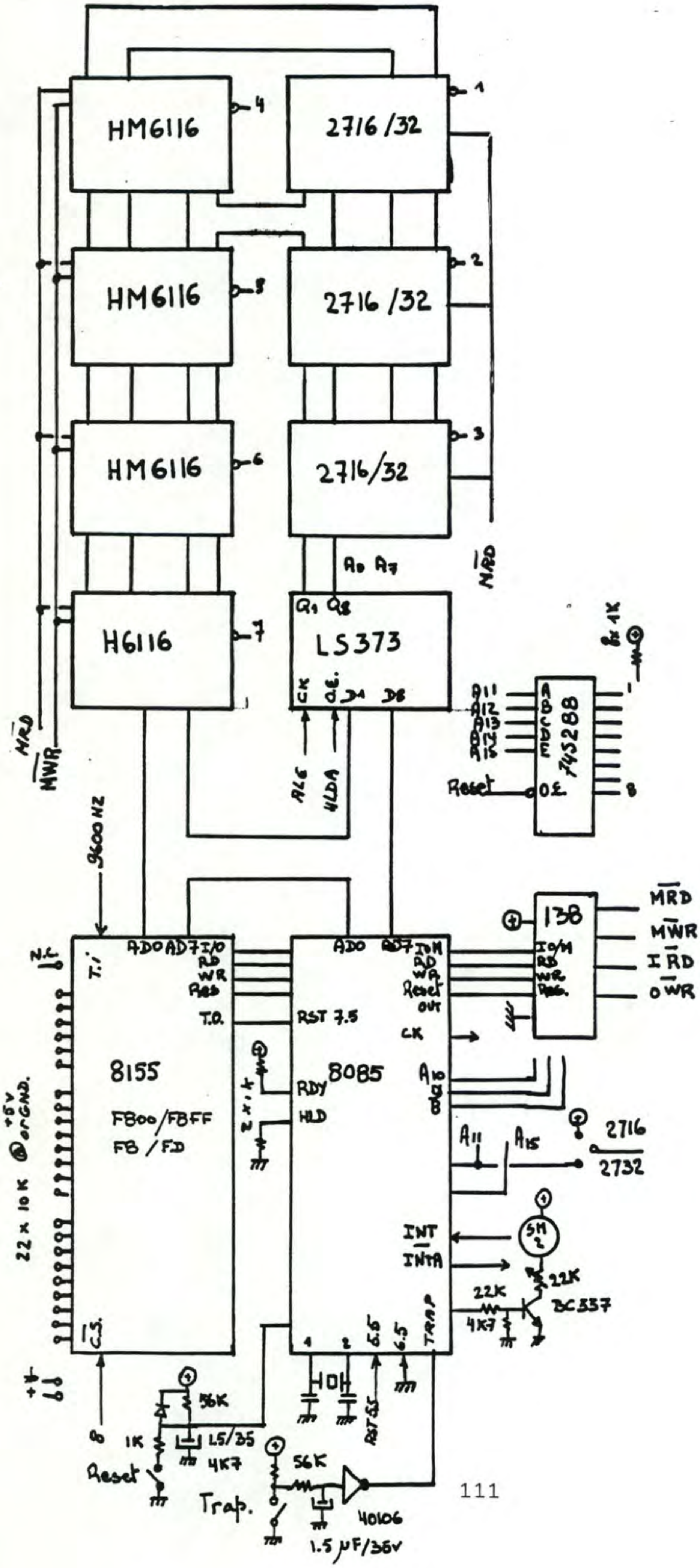
Pour des raisons de commodité, et compte tenu des contraintes exposées, l' adjonction de ces nouveaux circuits à la carte de base n' a put être retenue.

Il a été décidé le développement d' une nouvelle carte, destinée à supporter exclusivement la gestion du bus I.E.E.E.-488 selon le mode de contrôleur, ou encore selon le mode d' interfacage simple, en liaison avec une ligne de transmission de type V24.

Le schéma électronique de cette nouvelle carte est donnée ci-dessous:




Data
8291 OUT
8293 ...
8293 OUT
Con.
0
1
2
3
4
5
6
7
28
29
30
31
32
33
34
35
5
23
26
13
19
22
1
2
3
13
14
16
5
DAEQ
8291
CS
IRD
OWR
A0
A1
A2
RD
WR
RS0
RS1
RS2
Reset
D0
D7
REN
IFC
NRDF
NDAC
T/R2
DAV
EOI
SRQ
TI
INT
ATNO
IFCL
26 27
D0
D7
DAV
To IEEE 488 Con.
8291 + 92
8291 only.
3x.
74LS04
Res
CK
8292
Reset
DAV
COUNT
SRQ
IFC
REN
ATNO
IFCL
CIC
CLTN
SYC
EOI2
ATN1
IRD
OWR
A0
CS
D0
D7
TCI
SPI
OBF
IBF
5
4
Int.
27 26
SRQ
VRDF
NDAC
T/R2
NDAC
NRDF
SRQ
REN
IFC
ATN
EOI
To IEEE 488 Con.
4K7
(FAC)
SYSTEM Controller switch.
16
8259
IRD
1
2
3
4
8291 INT
TCI
SPI
OBF
IBF
8292
A0
CS
IRD
OWR
INT
To J4
Display
74C911
D0
D7
A0
A1
CS
OWR


HM6116
2716 /32
HM6116
2716 /32
HM6116
2716/32
H6116
LS373
MRD
MWR
A0 A7
Q1 Q8
CK
OE
D1 D8
ALE
HLDA
74S288
A11
A12
A13
A14
A15
Reset
OE
9600 HZ
8155
FB00/FBFF
FB / FD
AD0 AD7 I/O
RD
WR
Res
T.O.
T.I
C.S.
N.C
22 x 10 K
+5V or GND.
8085
AD0 AD7 IO/M
RD
WR
Reset OUT
RST 7.5
CK
RDY
HLD
A10
INT
INTA
TRAP
RST 5.5
6.5
138
MRD
MWR
I RD
O WR
2716
2732
A11
A15
22K
4K7
BC337
56K
1K
4K7
Reset
Trap.
56K
40106
1.5 μF/35V

2.4.1. Architecture matérielle et logicielle du contrôleur.

2.4.1.0. L' architecture matérielle.

Un chapitre précédent a permis d' établir la structure matérielle de la nouvelle carte appelée à supporter l' application de contrôleur IEEE-488.

Nous reprenons ci-dessous, l' architecture du contrôleur dans sa relation avec le monde extérieur, illustrée d' un point de vue plus fonctionnel (fig 2.4.1.0.).

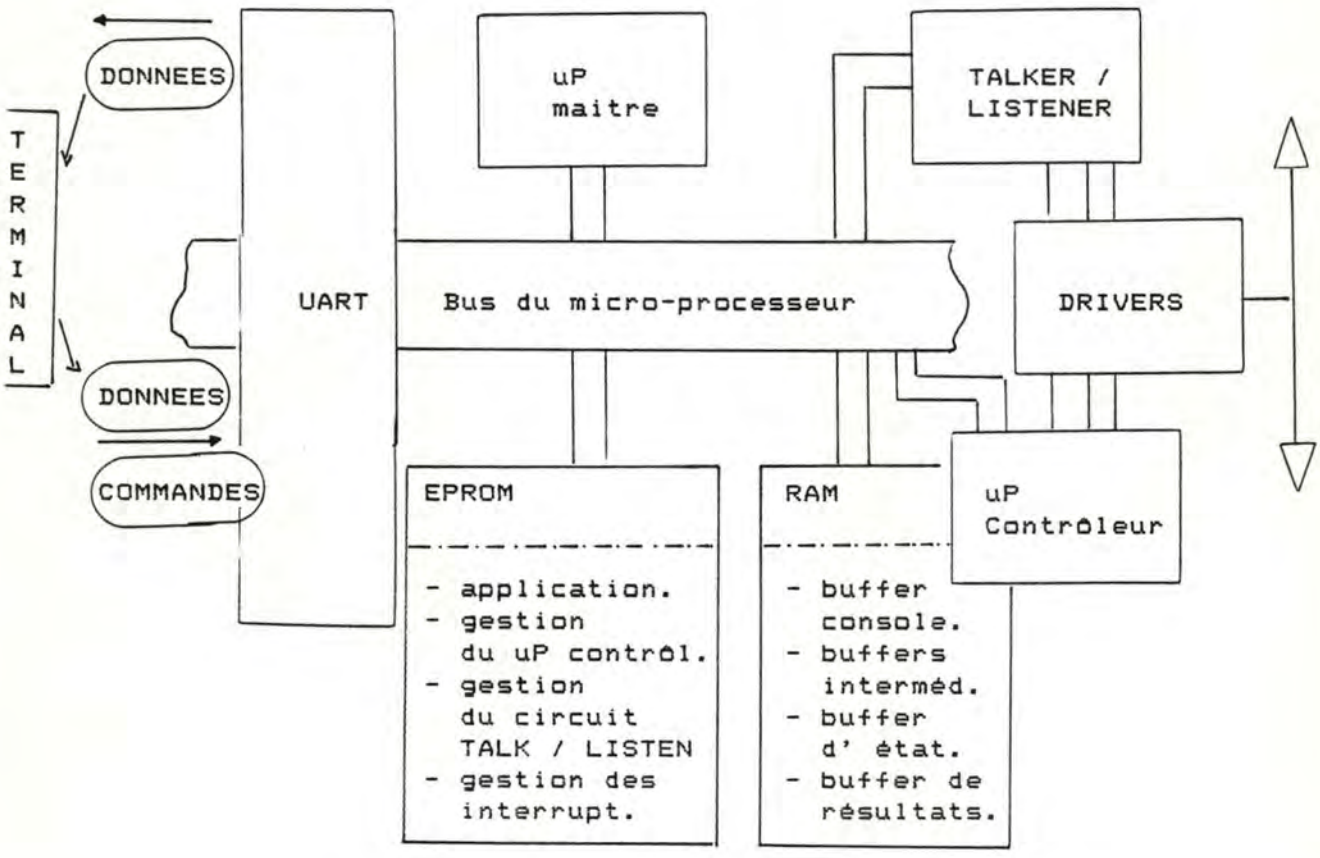


Fig 2.4.1.0. Structure d' interfacage du contrôleur en relation avec le monde extérieur.

Le contrôleur exécute les tâches de gestion et de contrôle du bus, à partir de commandes issues du micro-processeur principal (uP maitre).

Les programmes de gestion de l' application, résident en mémoire morte (EPROM), et réalisent les principales fonctions de gestion du bus IEEE-488, de mémorisation des résultats, de gestion des interruptions et de prise en compte des commandes issues d' un terminal ou de tout autre système situé à distance.

La mémoire vive (RAM), comprend les principaux buffers de console, de résultats ou de mémorisation temporaire, ainsi que les variables de travail.

Les fonctions de contrôle de gestion du transfert (notamment, le contrôle des séquences de synchronisation), sont exécutées au plus bas niveau de réalisation, par le micro-processeur contrôleur, qui possède ses propres programmes de gestion, ainsi qu' un banc de registres internes. Ce dernier travaille par ailleurs en étroite collaboration avec le circuit " TALKER-LISTENER ", qui réalise de manière quasi autonome, les fonctions de " SOURCE AND ACCEPTOR HANDSHAKE ".

La communication entre le contrôleur et l' utilisateur, s' éffectue à partir du schéma global suivant (fig. 2.4.1.1.) :

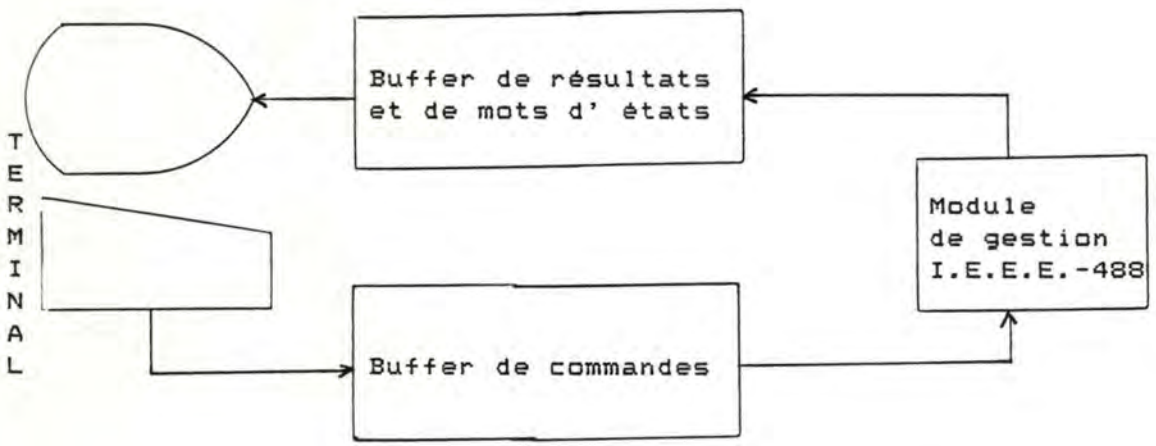


Fig. 2.4.1.1. Schéma de communication entre l' utilisateur et le contrôleur.

Le module de gestion I.E.E.E.-488, réalise les opérations de contrôle du bus, à partir des commandes soumises par l'utilisateur. Ces commandes donnent le plus souvent lieu, à une communication de résultats, ou d' informations d' états relatives à l' exécution en cours. Les buffers de commandes et de communication de résultats ou d' informations d' états, constituent, à ce titre, l' unique lien de communication entre le contrôleur et l' utilisateur.

Les commandes relèvent de deux types distincts :

- les commandes dont l' exécution est immédiate.

- les commandes dont l' exécution est postposée.

Les premières comprennent les demandes de communication d' informations d' état ou de suspension d' exécution en cours, les secondes sont relatives aux commandes aboutissant dans une file d' attente, et dont l' exécution est donnée explicitement au moyen d' un ordre de type " EXECUTE ", appartenant à la première catégorie.

Lors de l' exécution d' une liste de commandes de la deuxieme catégorie, seules les instructions de suspension d' exécution et de demande d' informations d' état sont alors prises en compte. Il y a dans ce cas précis, soit une suspension de l' activité opérée sur le bus, soit la transmission d' informations d' état, relatives à l' exécution en cours.

L'exécution de commandes donne habituellement lieu à un résultat. Ce résultat est construit et mémorisé pendant toute la durée d' exécution du module de gestion du bus, pour être ensuite transmis à l' utilisateur, par le biais du buffer de résultats.

Ce buffer s' articule autour d' une organisation circulaire, qui maximise l' utilisation de place libre, à l' intérieur d' une structure de taille fixe. Ce type d' organisation se prête par ailleurs bien, à la communication d' informations entre processus distincts que pourrait supporter l' application et qui devrait permettre d' exécuter en quasi-simultanéité, la saise de résultats et leur production auprès de l' utilisateur.

2.4.1.1. L' architecture logicielle.

Le logiciel relatif à l' application, s' articule autour de trois processus distincts, synchronisés par des indicateurs, et dont la communication se réalise le plus souvent par le biais de buffers d' informations.

Ces processus opèrent à l' intérieur d' un environement multi-tâches, ou chaque tâche procède à son propre blocage, dans l' attente d' un évenement spécifique, le plus souvent, et pour lesquelles la réactivation s' effectue par un processus superviseur, selon un procédé de tour de rôle.

Les trois tâches prennent ici le nom de " processeur ", afin de marquer le caractère d' indépendance qui les caractérise, ainsi que la volonté de faire abstraction de la machine physique appelée à les supporter.

Cette architecture multi-tâches répond aux contraintes relatives à la prise en compte en temps réel des évenements associés à la gestion du bus.

Cette approche permet ainsi de réaliser en quasi-simultanéité les fonctions de saisie de l' information en provenance d' un terminal, le gestion et le contrôle du bus I.E.E.E.-488, ainsi que les transferts de résultats.

La possibilité est ainsi donnée à l' utilisateur, de pouvoir s' informer sur l' évolution des traitements en cours, en introduisant une demande d' état (demande de status), ou encore, d' agir sur ces mêmes traitements, en donnant une commande de suspension d' exécution.

De même, les transferts de résultats ou d' informations d' états à destination du terminal, peuvent se réaliser parallèlement à l' exécution des fonctions de gestion et de contrôle du bus.

Une présentation succincte des trois " processeurs-logiciel ", est donnée ci-dessous.

2.4.1.1.0. Le processeur des messages d' entrée.

Le processeur des messages d' entrée, réalise les fonctions de réception et d' analyse des messages issus d' un terminal, aux fins de traitements ultérieurs (fig. 2.4.1.2.).

La prise en compte d' un message, s' éffectue caractère par caractère, à partir d' interruptions.

Les caractères font l' objet d' une première analyse, permettant de dissocier les commandes dont la prise en compte est immédiate (demande d' informations d' état ou de suspension d' exécution en cours), des commandes dont l' exécution est postposée.

Il est à noter, que seules les commandes dont l' exécution est immédiate peuvent être prise en compte à tout moment. Ces commandes bénéficient, vu leur caractère, d' un droit de pré-emption absolu sur le déroulement des opérations en cours.

Afin d' éviter une certaine lourdeur dans l' exécution des tâches menées par le micro-processeur maitre, et dans un soucis d' optimisation des transferts sur le bus I.E.E.E.-488, toute tentative d' introduction d' autres commandes lors d' une exécution en cours, aurait pour effet de suspendre l' exécution du processus d' entrée, jusqu' à la fin de cette exécution.

Une autre suspension du processeur des messages d' entrée, s' opère lors de son activation par le processeur moniteur en l' absence de caractères en entrée.

La réactivation de ce processus se réalise alors, par l' interruption que suscite la présence d' un nouveau caractère.

La disponibilité d' une ligne de commande dans le buffer de console, est signalée au processeur moniteur, dès la réception d' un caractère délimiteur de ligne.

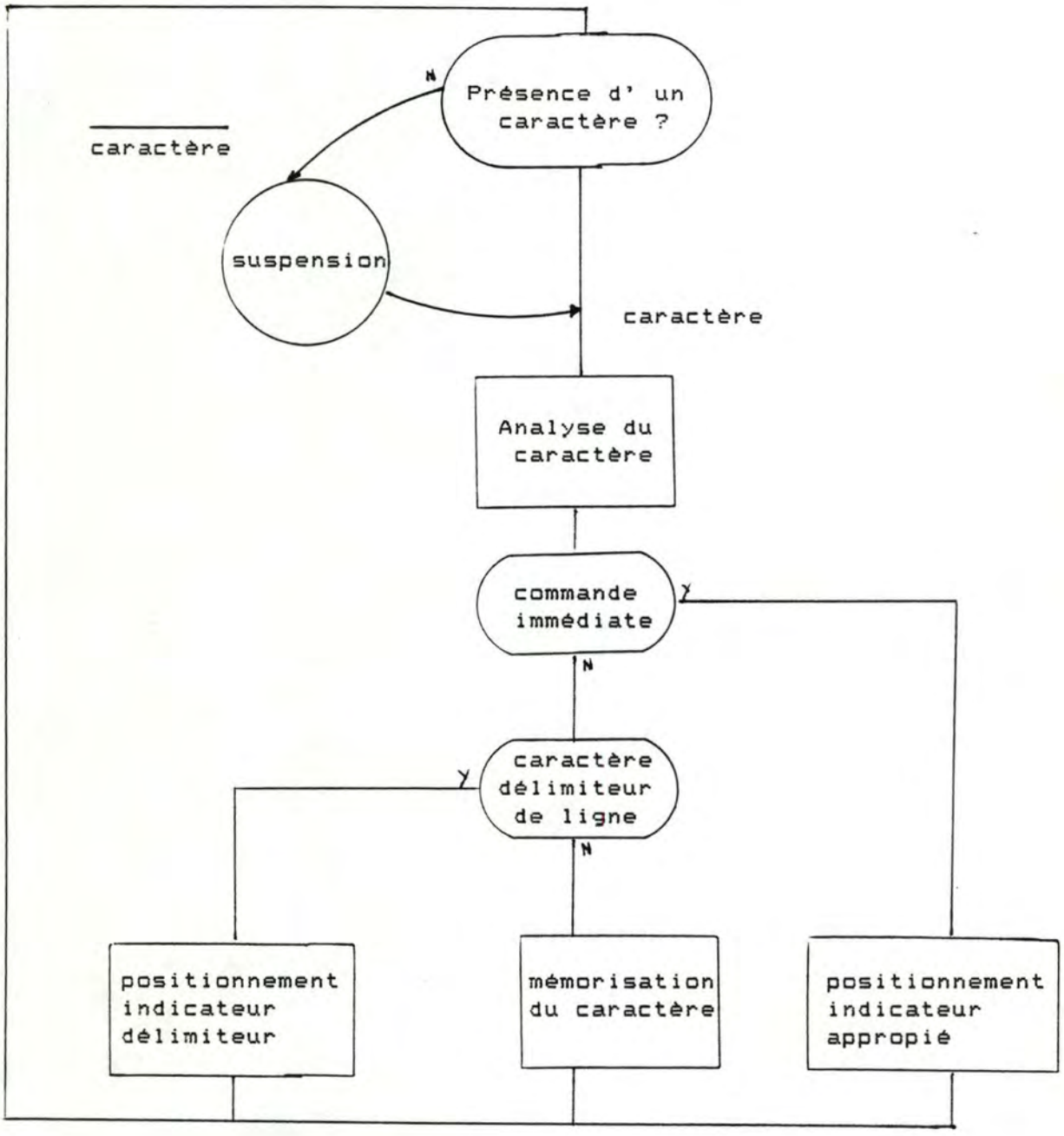


Fig. 2.4.1.2. Processeur des messages d' entrée.

2.4.1.1.1. Le processeur des messages de sortie.

Ce processeur réalise les fonctions de transfert des résultats ou d' informations d' état, à destination du terminal (fig. 2.4.1.3.).

L' activation du processus se réalise le plus souvent, à partir d' une prise en compte des interruptions signalant la disponibilité du circuit périphérique à transmettre un caractère, mais peut aussi s' éffectuer par le processeur moniteur lors d' une activité de bus réduite.

La principale relation avec un processus externe, se réalise par le biais du buffer de résultats. Ce buffer est organisé autour d' une structure circulaire, et recoit les informations issues des modules de gestion du bus. L'absence de résultats a pour effet de suspendre l' exécution de ce processus de transfert.

Deux autres formes de suspension peuvent se produire.

La première, est issue de l' état d' indisponibilité du terminal ou d' un organe périphérique lent, et signalé à l' interface contrôleur, par des caractères de contrôle de suspension et de réactivation (XOFF-XON). La suspension s' opère alors, jusqu' au moment de la réactivation que provoque le caractère de contrôle correspondant.

La seconde forme de suspension, se produit lors du transfert par le circuit périphérique, du caractère courant. Ceci provoque une indisponnibilité momentanée du circuit à recevoir un nouveau caractère. La réactivation s' éffectue alors par le processus superviseur, lors de la consultation du mot d' état associé au circuit, qui marque sa nouvelle disponibilité à éffectuer un transfert.

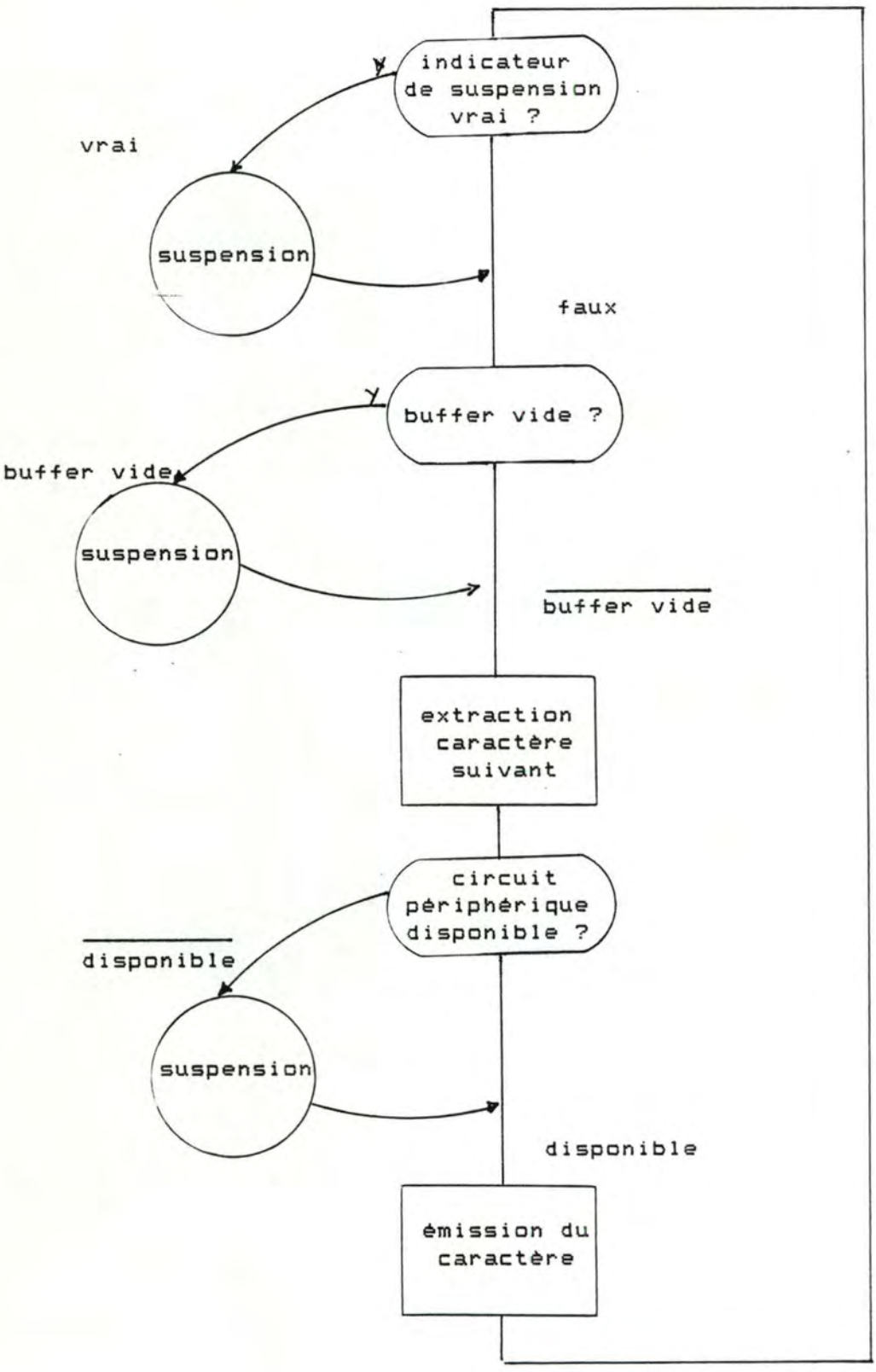


fig. 2.4.1.3. Processeur des messages de sortie.

2.4.1.1.2. Le processeur moniteur.

La principale tâche du processeur moniteur, consiste à prendre en compte les commandes en provenance du terminal, les exécuter, et transmettre les résultats à destination du terminal.

Pour ce faire, ce processus s' appuie, dans un premier temps, sur le processeur des messages d' entrée qui éffectue la mémorisation d' une ligne de commande. La disponibilité de cette ligne est signalée au processeur moniteur, par le biais d' un indicateur.

Le processeur moniteur invoque ensuite un module de contrôle syntaxique, éffectuant un contrôle de validité de la ligne de commande, ainsi que sa codification en l' absence d' erreur. Une erreur de syntaxe a pour effet d' interrompre le contrôle en cours, et de signaler à l' utilisateur le mot fautif. Toute nouvelle ligne de commande valide se substitue alors à la ligne erronée.

Une distinction s' opère alors, entre les commandes de type immédiate, et les commandes aboutissant dans une file d' attente, et dont l' exécution ultérieure sera à formuler explicitement.

Il s' agit dans le premier cas, de l' invocation du module d' exécution après la procédure de codification, et, dans le second, de la mémorisation sous forme codée, de la ligne de commande dans un buffer de commandes interprétées.

L' invocation du module d' exécution par le processeur moniteur, a pour effet de suspendre la prise en compte de commandes non-immédiates. Toute suspension réalisée à l' intérieur du module de gestion du bus, a pour effet de rendre temporairement le contrôle au processus superviseur, qui est alors libre d' activer le processeur des messages de sortie, afin de communiquer les résultats éventuels, ou le processeur des messages en entrée, afin de détecter d' éventuelles commandes immédiates.

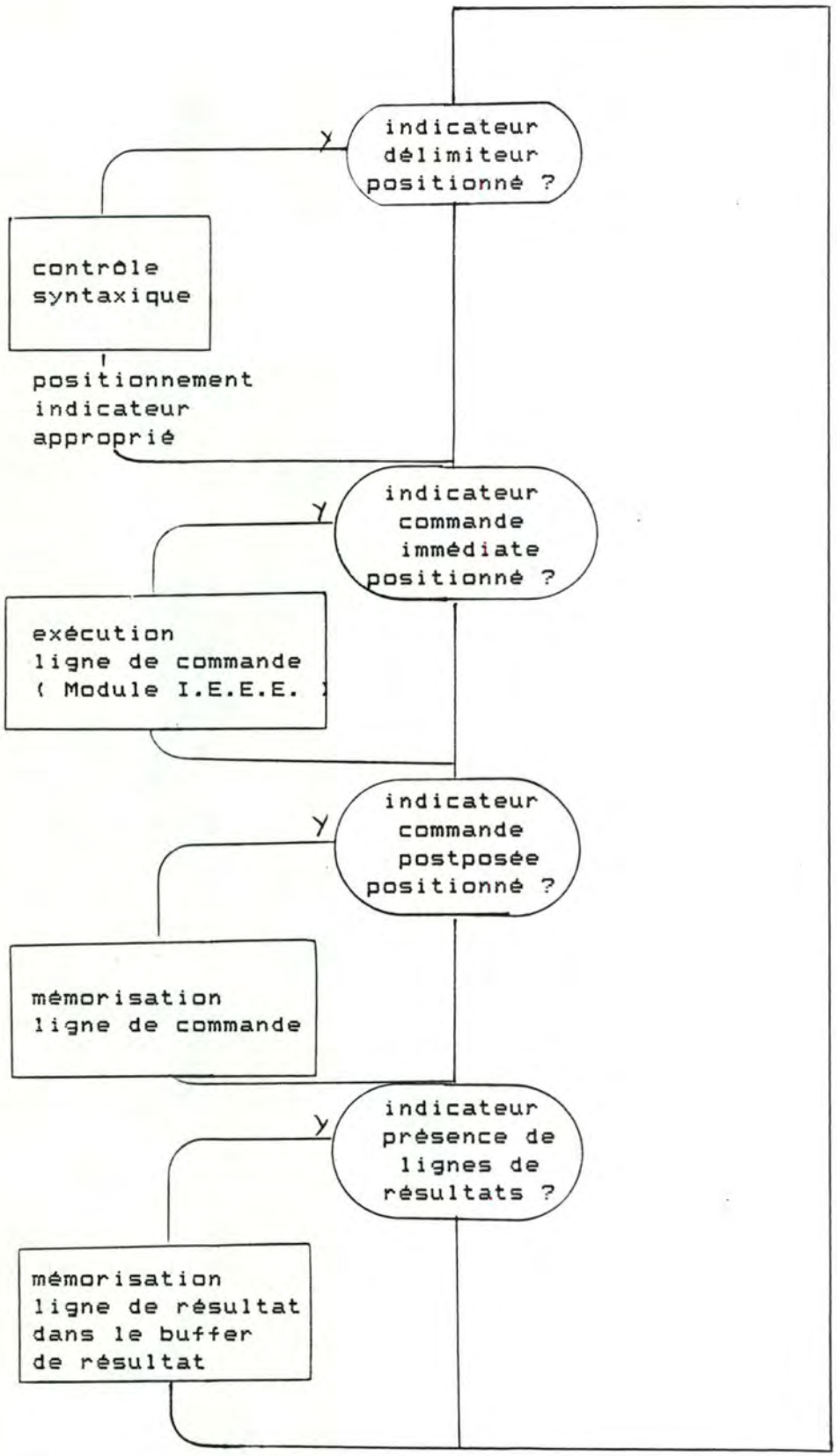


Fig. 2.4.1.4. Le processeur moniteur.

2.4.1.1.2. Demandes d' informations d' état.

Lors de l' exécution d' une séquence de commandes par le processeur-moniteur, la possibilité est donnée à l' utilisateur, de connaitre l' état de l' exécution en cours, par le biais d' un mot d' état.

Cette commande est prise en compte par le processeur des messages d' entrée, qui signale la requête au processeur moniteur. Ce dernier interrompt alors tout transfert éventuel de résultats vers le terminal, pour y substituer l' information d' état, présentée sous un format distinct.

Une fois l' information transmise, le processeur moniteur poursuit le transfert éventuel de résultats.

2.4.1.1.3. Suspension d' exécution en cours.

Cette commande permet à l' utilisateur d' agir sur l' exécution en cours, en suspendant toute activité de contrôle du bus.

Une fois la commande analysée par le processeur des messages d' entrée, sa signification est prise en compte par le processeur moniteur, qui tentera d' interrompre l' exécution du module de gestion du bus, lors de la prochaine suspension de ce processus.

Le processeur moniteur veillera alors, à placer les différents appareils de la configuration, dans un état cohérent, afin de pouvoir reprendre le contrôle ultérieurement.

2.5. Conclusions.

La réalisation du contrôleur de bus semble répondre aux contraintes de réalisations et aux objectifs fixés.

Des problèmes relatifs à l' utilisation du circuit contrôleur de bus (erreurs de programmation du circuit non renseignées), ont été rencontrés dans la phase de test du projet, et ont contribués à freiner considérablement la mise en exploitation du produit.

L' adoption de circuits spécialisés dans le développement de la carte électronique, constitue cependant un choix opportun.

Le développement d' autres formes d' interface I.E.E.E.-488, tels des adaptateurs d' appareils de mesure, des convertisseurs de protocole, des organes de saisie d' information ou de mémorisation, notamment, peuvent être envisagés sans aucune remise en cause de la carte électronique.

Dans le cas ou de nouveaux développements ne nécessiteraient le support que de l' une ou l' autre fonction d' interfacage, et ce, pour un nombre important d' exemplaires d' un même produit, il pourrait être envisagé d' appliquer les critères de réduction rencontrés, et, éventuellement, d' autres formes d' implémentation.

En adoptant la carte électronique de base, de nouveaux développements se limiteraient à l' aspect logiciel, et ne devrait nécessiter, compte tenu des routines et des procédures existantes, qu' un temps de programmation relativement restreint.

L' utilisation d' un compilateur 'C', utilisé actuellement pour de nombreux développements analogues, devrait pouvoir être envisagé lors de projets futurs, et notamment avec l' utilisation de micro-processeurs plus performants

Ce choix se caractérise par une plus grande souplesse en matière de développement, ainsi qu' une maitenance du logiciel plus aisée. Une réduction de performance associée à ce type de développement reste cependant à craindre, l' occupation en mémoire ne constituant plus aujourd' hui un critère déterminant.

Les développements relatifs au bus I.E.E.E.-488 constituent un atout pour l' avenir. L' apparition sans cesse croissante de produits incorporant l' interface à titre principal ou auxiliaire, incitent de nombreux utilisateurs à développer leur propre configuration d' appareils, et à sélectionner ces derniers en fonction de l' interface offerte.

Les interfaces d' adaptation au bus devraient, selon nous, suivrent cette évolution et permettre, grace à la souplesse du développement logiciel, de répondre aux exigences particulières des utilisateur.

Bibliographie.

ALLWORTH S.T., Introduction to Real-time Software Design, The Macmillan Press Ltd, London, 1981.

BASTIDE G., VELLAS J.R., Mise en oeuvre du bus I.E.E.E. 488 - utilisation et réalisation d' appareils, Editests, Paris 1982.

BROWN A. W., GLEAVES R. E., Modula-2: Pascal' s powerful heir, in Mini-micro Systems, September 1983, pp. 183-186.

BRUNIN J., Logique binaire des Circuits cablés et des Programmes enregistrés, Presse universitaires de Namur, Namur, 1975.

DEMERS C., Authoring a dedicated Operating System in Pascal, in Computer Design, April 1983, pp. 119-123.

DIETRICH H. E., Visualizing Interface Bus Activity, in Hewlett-Packard Journal, January 1974, pp. 19-23.

FINGERM., Problem Oriented Language, Part 2: Writing a Module, in Byte, January 1983, pp. 254-269.

GILBREATH J. and G., Eratosthenes Revisited, in Byte, January 1983, pp. 283-326.

HOLTER J., Add Multiple Tasks to Your Communication and Control Program, in Byte, September 1983, pp.445-478.

KLAISS D. E., Firmware Intelligence for Measurement and Control Processing, in Hewlett-Packard Journal, February 1978, pp. 10-18.

LEWIS J., Some Notes on Modular Assembly Programming, in Byte, December 1979, pp. 222-226.

OPPENHEIMER C. P., Reliable Designs Begins with the Basics, in Computer Design, August 1983, pp. 93-99.

PEASE S., Long- and Short-term Factors affect development-system Choise, in EDN, November 1983, pp. 259-270.

RANDLE W. C., KERTH N., Microprocessors in Instrumentation, in Proceedings of the I.E.E.E., February 1978, Vol 66 No 2, pp. 172-181.

RICCI D. W., STONE P. S., Putting Together Instrumentation Systems at Minimum Cost, in Hewlett-Packard Journal, March 1975, pp. 5-11.

RIFFKIN E. M., WILLIAMS S., The C Language : Key to Portability, in Computer Design, August 1983, pp. 143-150.

SCHULTZ S. E., TRIMBLE C. R., Filling in the Gaps - Modular Accessories for Instrument Systems, in Hewlett-Packard Journal, March 1975, pp. 12-18.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, I.E.E.E. Standard Digital Interface for Programmable Instrumentation (ANSI/IEEE Std 488-1978 - Revision of ANSI/IEEE Std 488-1975), New York, NY, 1978.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, I.E.E.E. Recommended Practices for Code and Format Conventions, New york, NY, 1982.

LONG J., Interfacing Microcomputers to Laboratory Instruments, in Microsystems, April 1983, pp. 62-69.

LOUGHRY D., A Common Digital Interface for Programmable Instruments: The Evolution of a System, in Hewlett-Packard Journal, November 1972, pp. 8-11.

LOUGHRY D., Viewpoints - Don Loughry on ANSI/IEEE Standard 488 and the HP Interface Bus, in Hewlett-Packard Journal, December 1979, pp. 27-28.

LOUGHRY D., The Hewlett-Packard Interface Bus: Current Perspectives, in Hewlett-Packard Journal, March 1975, pp. 2-4.

LOUGHRY D. C. , ALLEN M. S., I.E.E.E. Standard 488 and Microprocessor Synergism, in Proceedings of the I.E.E.E., February 1978, Vol 66 No 2, pp. 162-171.

NELSON G. E., RICCI D. W., A Practical Interface System for Electronic Instruments, in Hewlett-Packard Journal, November 1972, pp. 2-7.

Mc CORMACK J., Modula-2. A Worthy Successor to Pascal, in Byte, April 1983, pp. 385-394.

NEWROCK R., An IEEE-488 Bus Tutorial, in Microsystems, April 1983, pp. 34-61.

PARNAS D. L., A Technique for Software Module Specification with Examples, Communications of the ACM, May 1972, pp. 330-336.

PARNAS D. L., On the Criteria To Be Used in Decomposing Systems into Modules, Communications of the ACM, December 1972, pp. 1053-1058.

WYSS C. R., A Conceptual Approach to Real-Time Programming, in Byte, May 1983, pp. 452-470.

ZAKS R., LESEA A., Techniques d' Interfacage aux Microprocesseurs, Sybex, Paris, 1978.

Facultés Universitaires Notre-Dame de la Paix, Namur.
Institut d' Informatique.

Année Académique 1983-1984.

Conception et Réalisation d' un contrôleur de bus I.E.E.E.-488.

(Annexes)

Vanobberghen William.

Mémoire présenté en vue de l' obtention du grade de Licencié et Maitre en Informatique.

Promoteur de mémoire : Mr. J. Brunin.

Annexes.

Description des fonctions d' interfacage.

Les fonctions à remplir par l' interface sont au nombre de dix. Pour chacune d' entre elle nous trouverons :

- une description succincte de la fonction.

- une description de l' automate séquentiel susceptible de remplir la fonction dévolue.

Cette dernière description sera donnée au moyen d' un graphe, conformémant aux conventions de représentation associées à un automate séquentiel.

1. La fonction SH (Source Handshake).

Description succincte.

L' interelation qui existe entre cette fonction et une ou plusieurs fonctions de type AH (acceptor handshake), supportée chacune par un appareil distinct, permet de réaliser de manière asynchrone, un transfert de messages multilignes.

La fonction d' interfacage SH prend en charge le transfert d' un message multiligne d' un byte à partir du début et jusqu' à la fin du transfert, au moyen des lignes de synchronisation DAV, RFD et DAC.

Description de l' automate séquentiel.

L' implémentation de cette fonction devrait idéalement être réalisée à partir du diagramme d' état donné ci-après.

Les entrées de la fonction sont présentées sous forme de conditions primaires comprenant les messages suivants :

- messages externes : ceux-ci sont issus de l' état des lignes ATN, RFD et DAC de synchronisation.

- messages internes : ceux-ci comprennent les messages locaux de mise sous tension (pon), et de disponibilité d' une nouvelle donnée depuis la partie fonctionnelle de l' appareil (nba).

- états internes : ces états sont issus des autres modules de l' interface et sont symbolisés par "eif".

- signaux de temporisation : symbolisés par "T", ces signaux réalisent la transition selon les contraintes fixées par la norme.

Les mémoires du système sont constituées par les différents états que peut prendre la fonction, conformément à la réalisation des conditions primaires.

Les sorties du système sont constituées par les différentes actions que l' on associe à chacun des états du système. Par souci de clarté et afin de ne pas surcharger le graphe, nous les indiquerons dans un tableau séparé.

Ces actions sont principalement liées au contrôle de la ligne DAV par la fonction SH, et de l' interaction existant avec la partie fonctionnelle de l' appareil.

2. Entrées (Conditions Primaires).

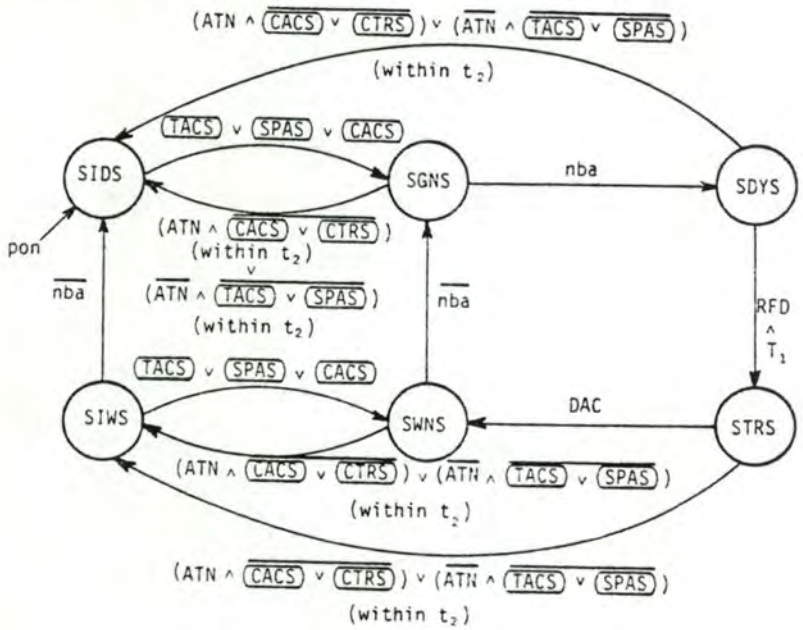
SH Mnemonics

Messages		Interface States	
t: pon	= power on	SIDS	= source idle state
t: nba	= new byte available	SGNS	= source generate state
Bc: ATN	= attention	SDYS	= source delay state
Bh: RFD	= ready for data	STRS	= source transfer state
Bh: DAC	= data accepted	SWNS	= source wait for new cycle state
		SIWS	= source idle wait state

eif:
- $\overline{TACS}$ = talker active state (T function)
- $\overline{SPAS}$ = serial poll active state (T function)
- $\overline{CACS}$ = controller active state (C function)
- $\overline{CTRS}$ = controller transfer state (C function)

1. Mémoires d' états

SH State Diagram



3. Sorties.

SH Message Outputs

SH State	Remote Message Sent DAV	Device Function (DF) Interaction
SIDS	(F)	DF can change remote multiline messages
SGNS	F	DF can change remote multiline messages
SDYS	F	DAB, EOS multiline, and END messages shall not change
STRS	T	DAB, EOS multiline, and END messages shall not change
SWNS	T or F	DF requested to change multiline messages
SIWS	(F)	DF requested to change multiline messages

2. La fonction AH (Acceptor Handshake).

Description succincte.

Cette fonction assure à un appareil, la bonne réception d' un message de type multiligne par le biais de la séquence de synchronisation, en invoquant les fonctions SH et AH.

La possibilité est donnée de pouvoir retarder le début ou la fin d' un transfert, tout en étant assuré de pouvoir le poursuivre à tout moment.

Cette fonction utilise les messages DAV, RFD et DAC lors du transfert de chaque byte.

Description de l' automate séquentiel.

Le diagramme d' état présenté à la page suivante, donne les conditions d' implémentation de la fonction AH.

Les entrées de la fonction sont présentées sous forme de conditions primaires qui comprend les messages ou les état suivants :

- messages externes : ceux-ci sont relatifs à l' état des lignes ATN et DAV des bus de contrôle et de synchronisation, respectivement.

- messages internes : ces messages sont constitués des messages locaux de mise sous tension (pon), de disponibilité à recevoir un nouveau message (rdy) et la prise de contrôle de manière synchrone (tcs).

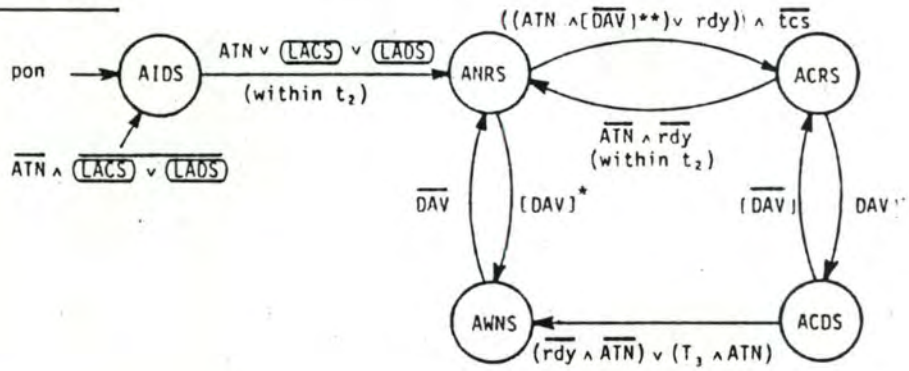
- états internes : ces états, repris sous le symbole "eif", sont relatifs à l' exitence d' un adressage en mode LISTENER (LADS) et d' un état actif de LISTENER (LACS).

- signaux de temporisation : ceux-ci respectent les contraintes fixées par la norme et données sous la forme d' un tableau que nous reprenons dans ces annexes.

Les mémoires du système, sont symbolisées au moyen des différents états que peut prendre la fonction.

1. Mémoires d' états

AH State Diagram



2. Entrées (Conditions Primaires).

AH Mnemonics

	Messages		Interface States	
t	pon	= power on	AIDS	= acceptor idle state
	rdy	= ready for next message	ANRS	= acceptor not ready state
	tcs	= take control synchronously*	ACRS	= acceptor ready state
Bc	ATN	= attention	ACDS	= accept data state
Bhs	DAV	= data valid	AWNS	= acceptor wait for new cycle state
eif	LADS	= listener addressed state (L function)		
	LACS	= listener active state (L function)		

3. Sorties.

AH Message Outputs

AH State	Remote Message Sent RFD	Remote Message Sent DAC	Device Function (DF) Interaction
AIDS	(T)	(T)	DF cannot receive remote multiline or END messages
ANRS	F	F	DF cannot receive remote multiline or END messages
ACRS	(T)	F	DF cannot receive remote multiline or END messages
AWNS	F	(T)	DF cannot receive remote multiline or END messages
ACDS	F	F	DF can receive remote multiline or END messages if LACS is active

Signalons cependant que ces actions sont principalement liées à la gestion des lignes RFD et DAC du bus de synchronisation, ainsi qu' à l' interaction existant avec la partie fonctionnelle de l' appareil.

3. La fonction T (Talker).

Description succincte.

Cette fonction d' interfacage permet à un appareil d' émettre des informations liées au fonctionnement de l' appareil, par le biais de l' interface et à destination de plusieurs autres appareils.

Cette fonction se réalise grâce à un adressage préalable portant sur une ou deux adresses successives.

Dans le cas d' un adressage étendu, la fonction prendra la dénomination de TE (extended talker), signifiant que deux adresses successives sont nécessaires à la réalisation de la fonction.

Une seule forme d' adressage ne peut cependant être supportée au sein d' un même appareil.

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires comprenant les messages et états suivants :

- messages externes : ceux-ci sont issus de l' état des lignes IFC et ATN du bus de contrôle, ainsi que des contenus de messages multilignes MTA (My Talk Address), SPE (Serial Poll Enable), SPD (Serial Poll Disable), OTA (Other Talk Address), ainsi que MLA (My Listen Address).

- messages internes : ceux-ci sont issus des messages locaux de mise sous tension (pon), et du mode "talk only" (ton).

- états internes : ces états sont issus des autres modules de l' interface, et sont symbolisés par "eif". Signalons que l' état ACDS est issu de la fonction AH rencontrée plus haut, et que l' état SPMS (Serial Poll Mode) est activé lors de la réception du message SPE.

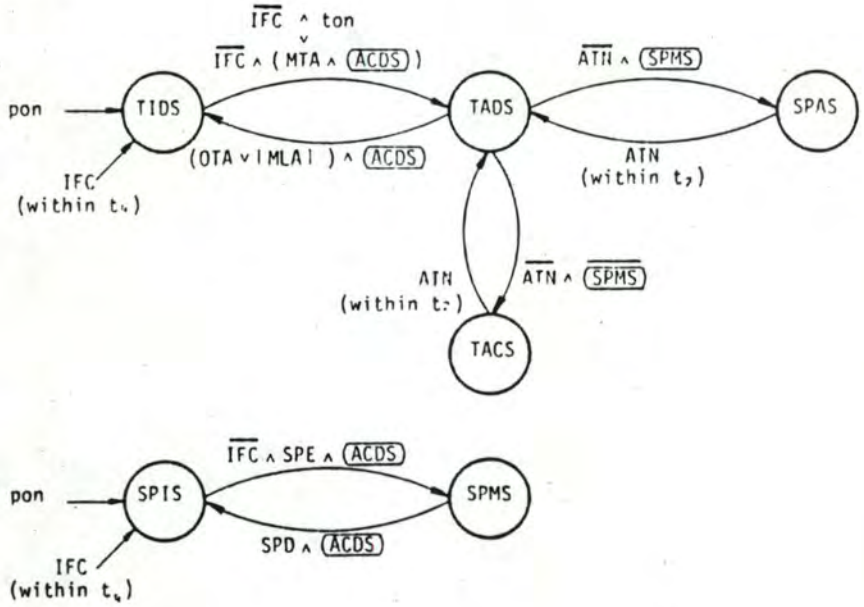
Les mémoires du système, symbolisées par les différents états que peut prendre la fonction, sont reprises dans le graphe des mémoires d' états.

Les sorties du système sont constituées par les actions associées à chacun des états, et relatives notamment à l' émission de messages.

Ces actions sont étroitement liées au contenu des messages distants de type uniligne ou multiligne, ainsi qu' à l' interaction avec la partie fonctionnelle de l' appareil.

1. Mémoires d' états

T State Diagram



2. Entrées (Conditions Primaires).

T Mnemonics

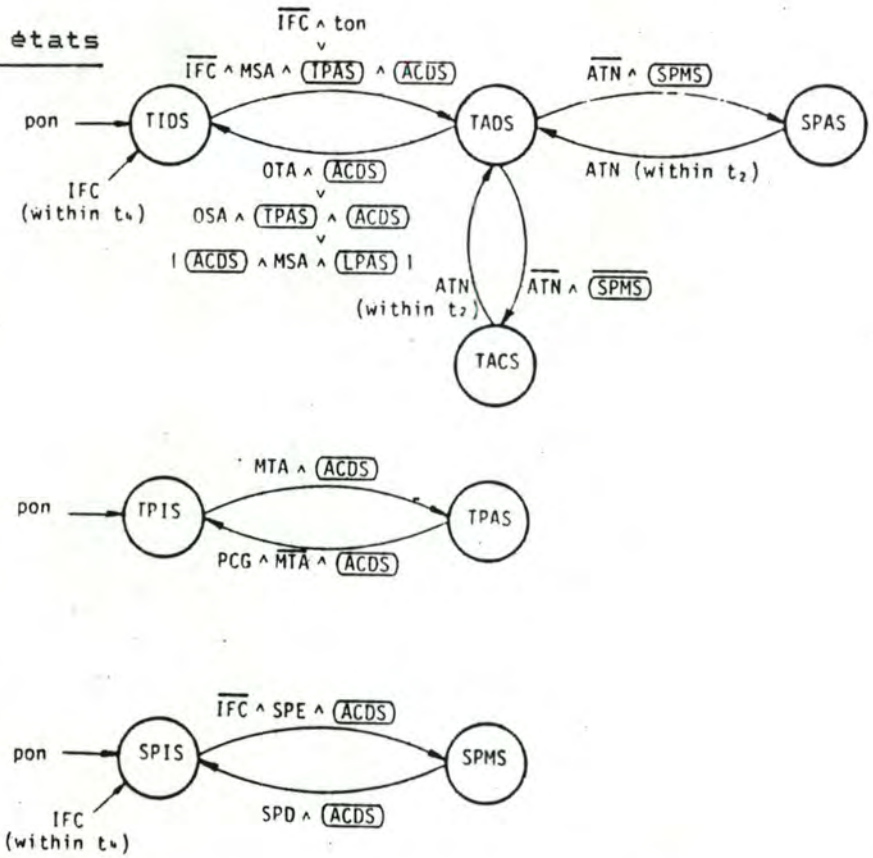
	Messages		Interface States	
t	pon	= power on	TIDS	= talker idle state
	ton	= talk only	TADS	= talker addressed state
Bc	IFC	= interface clear	TACS	= talker active state
	ATN	= attention	SPAS	= serial poll active state
m	MTA	= my talk address	SPIS	= serial poll idle state
	SPE	= serial poll enable	SPMS	= serial poll mode state
	SPD	= serial poll disable		
	OTA	= other talk address		
	MLA	= my listen address		
eif	ACDS	= accept data state (AH function)		

3. Sorties.

T or TE Message Outputs

T State	Qualifier	Remote Messages Sent*			Device Function (DF) Interaction
		Multiline	END	RQS‖	
TIDS		(NUL)	(F)	(F)	DF not allowed to send messages
TADS		(NUL)	(F)	(F)	DF not allowed to send messages
TACS		DAB† or EOS†	T or F†	(F)	DF can send DAB, EOS, or END message (if used) concurrent with DAB†
SPAS	APRS inactive	STB†	F or T	F	DF can send one STB message†
SPAS	APRS active	STB†	F or T	T	DF can send one STB message†

1. Mémoires d' états



NOTE: If the TE function is used together with the L function instead of the LE function, then [MSA ∧ (ACDS) ∧ (LPAS)] shall be replaced by [MLA ∧ (ACDS)].

TE State Diagram

2. Entrées (Conditions Primaires).

TE Mnemonics

	Messages		Interface States	
t	pon	= power on	TIDS	= talker idle state
	ton	= talk only	TADS	= talker addressed state
Bc	IFC	= interface clear	TACS	= talker active state
	ATN	= attention	SPAS	= serial poll active state
m	MTA	= my talk address	TPIS	= talker primary idle state
	OTA	= other talk address	TPAS	= talker primary addressed state
	OSA	= other secondary address	SPIS	= serial poll idle state
	PCG	= primary command group	SPMS	= serial poll mode state
	SPE	= serial poll enable		
	SPD	= serial poll disable		
	MSA	= my secondary address		
eif	(ACDS)	= accept data state (AH function)		
	(LPAS)	= listener primary addressed state (L function)		

4. La fonction L (Listener).

Description succincte.

Cette fonction permet à un appareil de recevoir des informations liées au fonctionnement de l' appareil depuis le bus d' interfacage, et en provenance de un ou plusieurs autres appareils. Cette fonction se réalise à partir d' un adressage préalable.

l' adressage s' éffectue habituellement à partir d' une adresse donnée en un byte. Un adressage étendu portant sur deux adresses successives d' un byte chacune est cependant possible. La fonction prend alors la dénomination de LE (extended listener).

Une seule forme d' adressage ne peut cependant être supportée au sein d' un même appareil.

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires, comprenant les messages et états suivants :

- messages externes : ceux-ci sont constitués par l' état des lignes IFC et ATN du bus de contrôle, et du contenu des messages multilignes UNL (Unlisten), MLA (My Listen Address), et MTA (My Talk Address).

- messages internes : ces messages sont relatifs à des messages locaux de mise sous tension (pon), d' écoute simple (ltn), exclusive (lon), ou d' inhibition de l' écoute en mode local (lun).

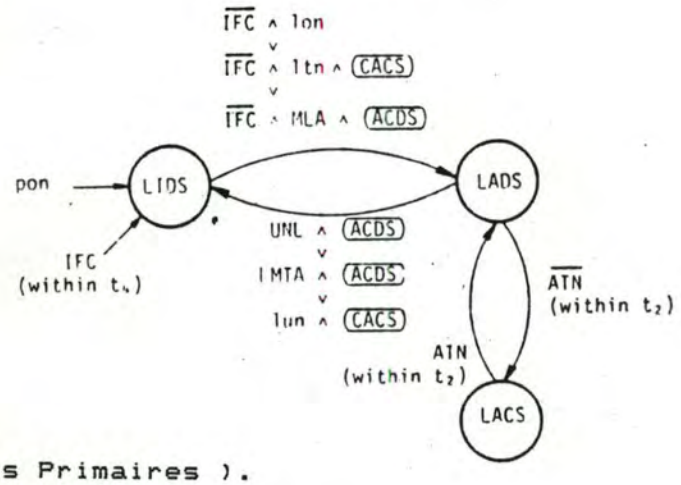
- états internes : ces états sont issus des autres modules de l' interface et sont symbolisés par "eif". Les états susceptibles de provoquer une transition sont l' état ACDS, issu de la fonction AH rencontrée, et l' état CACS (Controller Active State) associe à la fonction de contrôle C.

Les états LIDS, LADS et LACS constituent les principales mémoires du système.

Les actions associées à ces états, et qui constituent les sorties du système, sont relatives à l' interaction existant avec la partie fonctionnelle de l' appareil exclusivement.

1. Mémoires d' états

L State Diagram



2. Entrées (Conditions Primaires).

L Mnemonics

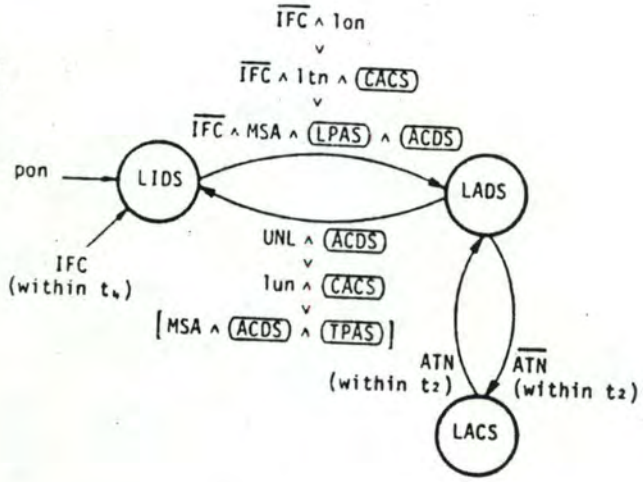
	Messages		Interface States	
t	pon	= power on	LIDS	= listener idle state
	ltn	= listen	LADS	= listener addressed state
	lun	= local unlisten	LACS	= listener active state
	lon	= listen only		
Bc	IFC	= interface clear		
	ATN	= attention		
m	UNL	= unlisten		
	MLA	= my listen address		
	MTA	= my talk address		
eif	(ACDS)	= accept data state (AH function)		
	(CACS)	= controller active state (C function)		

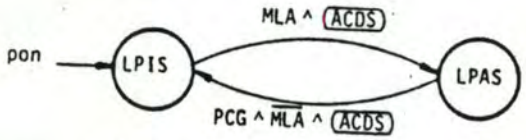
3. Sorties.

L or LE Message Outputs

L or LE State	Remote Messages Sent	Device Function (DF) Interaction
LIDS	none	DF not allowed to receive messages
LADS	none	DF not allowed to receive messages
LACS	none	DF can receive one device dependent message byte each time ACDS is active

1. Mémoires d' états





NOTE: If the LE function is used together with the T function, then [MSA ∧ (ADCS) ∧ (TPAS)] shall be replaced by [MTA ∧ (ADCS)].

LE State Diagram

2. Entrées (Conditions Primaires).

LE Mnemonics

Messages		Interface States	
pon	= power on	LIDS	= listener idle state
ltn	= listen	LACS	= listener active state
lun	= local unlisten	LADS	= listener addressed state
lon	= listen only	LPIS	= listener primary idle state
IFC	= interface clear	LPAS	= listener primary addressed state
ATN	= attention		
UNL	= unlisten		
MLA	= my listen address		
PCG	= primary command group		
MSA	= my secondary address		

(ACDS) = accept data state (AH function)
(CACS) = controller active state (C function)
(TPAS) = talker primary addressed state (T function)

5. La fonction SR (Service Request).

Description succincte.

Cette fonction permet à un appareil de solliciter l' attention du contrôleur, afin de pouvoir réaliser une action spécifique.

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires comprenant les messages et états suivants:

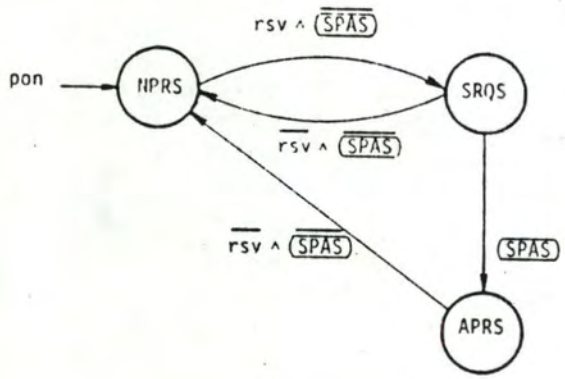
- messages internes : ces messages prennent la forme de messages locaux issus de la partie fonctionnelle de l' appareil. Les messages de mise sous tension (pon) et de demande d' interruption (rsv) constituent les seuls messages en entrée.

- états internes : l' état SPAS est issu de la fonction TALKER de l' appareil.

Les états NPRS, SRQS et APRS, constituent les mémoires du système.

Le message de sortie est relatif à la gestion de la ligne SRQ et aucune forme d' interaction avec la partie fonctionnelle de l' appareil n' est à envisager.

1. Mémoires d' états



SR State Diagram

2. Entrées (Conditions Primaires).

SR Mnemonics

Messages		Interface States	
t	pon = power on	NPRS	= negative poll response state
	rsv = request service	SRQS	= service request state
		APRS	= affirmative poll response state
eif	$\overline{SPAS}$	= serial poll active state (T function)	

3. Sorties.

SR Message Outputs

SRQ State	Remote Message Sent: SRQ	Device Function Interaction
NPRS	(F)	none
SRQS	T	none
APRS	(F)	none

6. La fonction RL (Remote Local).

Description succincte.

Cette fonction permet à un appareil de choisir parmi deux formes distinctes d' introduction de données : le panneau avant de l' appareil, ou l' interface.

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires comprenant les messages suivants :

- messages externes : il s' agit du message uniligne REN issu du bus de contrôle, ainsi que des messages multilignes LLO, GTL et MLA.

- messages internes : ceux-ci comprennent les messages locaux de mise sous tension (pon), ainsi que de retour en mode local (rtl).

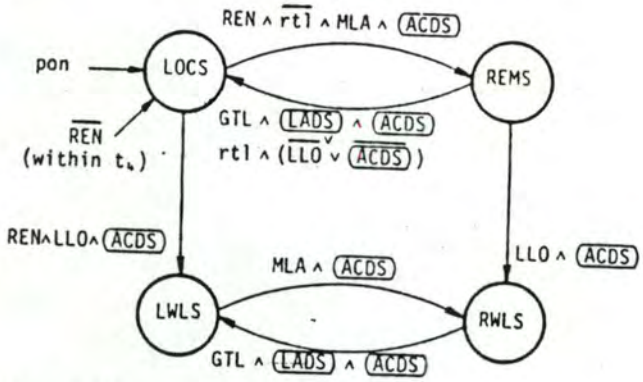
- états internes : ces états sont issus de deux autres fonctions de l' appareil : la fonction AH (message ACDS), et la fonction L (message LADS).

Les mémoires de la fonction sont constituées par quatre états : les états LOCS (Local State), LWLS (Local With Lockout State), REMS (Remote State), et RWLS (Remote With Lockout State).

les sorties du système se caractérisent par l' interaction de la fonction avec la partie fonctionnelle de l' appareil. L' appareil entrera dans l' un des deux modes d' introduction de données, suivant l' état qui aura été activé.

1. Mémoires d' états

RL State Diagram



NOTE: If the RL function is used together with the LE function, then the term MLA shall be replaced by the term (MSA ∧ (LPAS)).

2. Entrées (Conditions Primaires).

RL Mnemonics

	Messages		Interface States	
t	pon	= power on	LOCS	= local state
	rtl	= return to local	LWLS	= local with lockout state
bc	REN	= remote enable	REMS	= remote state
	LLO	= local lockout	RWLS	= remote with lockout state
m	GTL	= go to local		
	MLA	= my listen address		
eif	(ACDS)	= accept data state (AH)		
	(LADS)	= listener addressed state (L)		

3. Sorties.

RL Message Outputs

RL State	Remote Messages Sent	Device Function Interaction
LOCS	none	device is in "local control" mode
LWLS	none	device is in "local control" mode
REMS	none	device is in "remote control" mode
RWLS	none	device is in "remote control" mode

7. La fonction PP (Parallel Poll).

Description succincte.

La fonction PP permet à un appareil d' adresser au contrôleur le message PPR, sans avoir été adressé en TALKER au préalable.

Les lignes DIO1 à DIO8 sont utilisées pour véhiculer les bits d' états, lors d' une scrutation d' interruptions menée en mode parallèle.

Afin d' éviter tout conflit dans l' émission du bit d' état en concurrence avec d' autres appareils, chaque appareil supportant cette fonction, se voit assigner une ligne d' état unique qu' il utilisera lors de la scrutation parallèle menée par le contrôleur.

Cette faculté permet à huit appareils, au plus, d' être scrutés de cette manière, quoique le partage de même lignes par plusieurs appareils ôte toute limitation de ce nombre.

Cette procédure est à différencier de la procédure de scrutation selon le mode série, ou chaque appareil à tour de rôle fait l' objet d' une scrutation par le contrôleur.

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous la forme de conditions primaires comprenant les messages suivants :

- messages externes : il s' agit des messages ATN et IDY issus du bus de contrôle, ainsi que des messages multilignes PPE (Parallel Poll Enable), PPD (Parallel Poll Disable), PPC (Parallel Poll Configure), PCG (Primary Command Group) et PPU (Parallel Poll Unconfigure). Ces messages emanent du contrôleur exclusisvement.

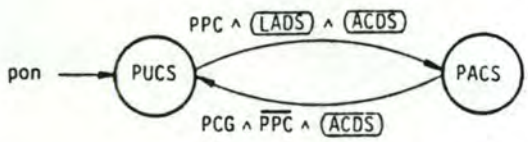
- messages internes : ceux-ci sont constitués par les messages de mise sous tension (pon), du mot d' état individuel (ist), ainsi que de la scrutation locale (lpe).

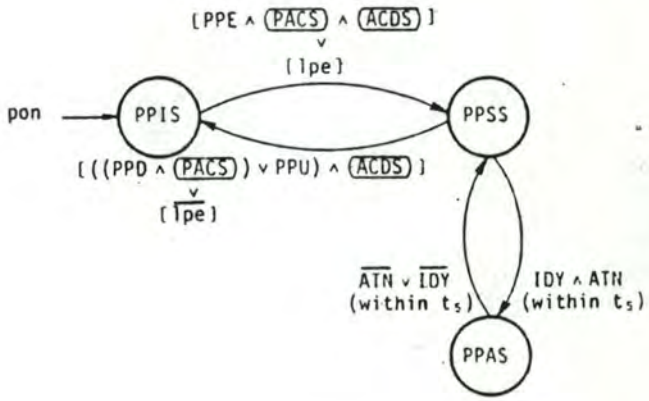
- états internes : ces états sont issus des fonctions AH (état ACDS) et L (état LADS), de l' appareil.

Les mémoires du syteme sont constituées des états PPIS (Parallel Poll Idle State), PPSS (Parallel Poll Standby State), PPAS (Parallel Poll Active State), PUCS (Parallel Poll Unadressed to Configure State) et PACS (Parallel Poll Adressed to Configure State).

La sortie du système est constituée par le message PPR, sans interaction avec la partie fonctionnelle de l' appareil.

1. Mémoires d' états





2. Entrées (Conditions Primaires).

PP Mnemonics

		Messages	Interface States
t	pon	= power on	PPIS = parallel poll idle state
	ist	= individual status (Table 25)	PPSS = parallel poll standby state
	lpe	= local poll enabled	PPAS = parallel poll active state
Bc	ATN	= attention	PUCS = parallel poll unaddressed to configure state
	IDY	= identify	PACS = parallel poll addressed to configure state
m	PPE	= parallel poll enable	
	PPD	= parallel poll disable	
	PPC	= parallel poll configure	
	PCG	= primary command group	
	PPU	= parallel poll unconfigure	
eif	(ACDS)	= accept data state (AH function)	
	(LADS)	= listener addressed state (L function)	

3. Sorties.

PP Message Outputs

PP State	Qualifier	Remote Message Sent PPRn*†	Device Function Interfaction
PPIS		(F)	none
PPSS		(F)	none
PPAS	ist ≡ S†	T	none
PPAS	ist ≠ S†	(F)	none

PPR Message Specified by Each of the Combinations of Values P1 Through P3

Bits Received with Most Recent PPE Command			PPR Message Specified
P3	P2	P1	
0	0	0	PPR1
0	0	1	PPR2
0	1	0	PPR3
0	1	1	PPR4
1	0	0	PPR5
1	0	1	PPR6
1	1	0	PPR7
1	1	1	PPR8

8. La fonction DC (Device clear).

Description succincte.

Cette fonction permet à un appareil de réaliser une procédure d' initialisation, à la suite d' une commande individuelle (adressée), ou groupée (non-adressée).

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires, comprenant les messages et états suivants :

- messages externes : ces messages sont constitués des commandes multilignes DCL (Device Clear), et SDC (Selected Device Clear).

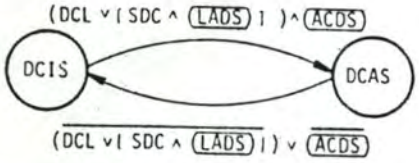
- états internes : ces états sont issus des fonctions AH (état ACDS) et L (état LADS) de l' appareil.

Les mémoires du système sont constituées des états DCIS (Device clear Idle State) et DCAS (Device clear Active State).

L' action associée au dernier état est liée à la partie fonctionnelle de l' appareil et consiste à entreprendre une procédure d' initialisation. En dehors de cet état spécifique, l' appareil poursuit normallement ses opérations.

1. Mémoires d' états

DC State Diagram



2. Entrées (Conditions Primaires).

DC Mnemonics

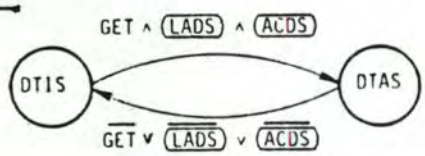
	Messages		Interface States
m	DCL = device clear	DCIS	= device clear idle state
	SDC = selected device clear	DCAS	= device clear active state
e i f	(ACDS) = accept data state (AH function)		
	(LADS) = listener addressed state (L function)		

3. Sorties.

DC Message Outputs

DC State	Remote Message Sent	Device Function (DF) Interaction
DCIS	none	normal device function operation
DCAS	none	DF should return to a known fixed state

1. Mémoires d' états



DT State Diagram

DT Mnemonics

	Messages		Interface States
m	GET = group execute trigger	DTIS	= device trigger idle state
		DTAS	= device trigger active state
e i f	(ACDS) = accept data state (AH function)		
	(LADS) = listener addressed state (L function)		

3. Sorties.

DT Message Outputs

DT State	Remote Messages Sent	Device Function (DF) Interaction
DTIS	none	normal DF operation
DTAS	none	DF should start performing triggered operation

9. La fonction DT (Device Trigger).

Description succincte.

Cette fonction permet à un appareil de réaliser une opération spécifique, à la suite d' une commande individuelle (commande adressée) ou groupée (commande non-adressée).

Description de l' automate séquentiel.

Les entrées de la fonction sont présentées sous forme de conditions primaires comprenant les messages et états suivants :

- message externe : il s' agit du message multiligne GET (Group Execute Trigger).

- états internes : ces états sont issus des fonctions AH (état ACDS) et L (état LADS) de l' appareil.

Les mémoires du système sont constituées des états DTIS (Device Trigger Idle State) et DTAS (Device Trigger Active State).

L' action associée au dernier état est liée à la partie fonctionnelle de l' appareil, et consiste à entreprendre la procédure d' activation de mesure, ou tout autre opération associée à cet état. En dehors de cet état spécifique, l' appareil poursuit le déroulement normal de ses opérations.

(SENT AND RECEIVED WITH ATN=1)

b_4	b_3	b_2	b_1	ROW	COLUMN 0	MSG (1)	COLUMN 1	MSG	COLUMN 2	MSG	COLUMN 3	MSG	COLUMN 4	MSG	COLUMN 5	MSG	COLUMN 6	MSG	COLUMN 7	MSG
0	0	0	0	0	NUL		DLE		SP		0		@		P				p	
0	0	0	1	1	SOH	GTL	DC1	LLO	!		1		A		Q		a		q	
0	0	1	0	2	STX		DC2		"		2		B		R		b		r	
0	0	1	1	3	ETX		DC3		#		3		C		S		c		s	
0	1	0	0	4	EOT	SDC	DC4	DCL	$		4		D		T		d		t	
0	1	0	1	5	ENQ	PPC (1)	NAK	PPU	%	MLA ASSIGNED TO DEVICE	5	MLA ASSIGNED TO DEVICE	E	MTA ASSIGNED TO DEVICE	U	MTA ASSIGNED TO DEVICE	e	MEANING DEFINED BY PCG CODE	u	MEANING DEFINED BY PCG CODE
0	1	1	0	6	ACK		SYN		&		6		F		V		f		v	
0	1	1	1	7	BEL		ETB		'		7		G		W		g		w	
1	0	0	0	8	BS	GET	CAN	SPE	(		8		H		X		h		x	
1	0	0	1	9	HT	TCT	EM	SPD	)		9		I		Y		i		y	
1	0	1	0	10	LF		SUB		*		:		J		Z		j		z	
1	0	1	1	11	VT		ESC		+		;		K		[		k		{	
1	1	0	0	12	FF		FS		,		<		L		\		l		|	
1	1	0	1	13	CR		GS		-		=		M		]		m		}	
1	1	1	0	14	SO		RS		.		>		N		^		n		~	
1	1	1	1	15	SI		US		/		?	UNL	O	UNT	_		o		DEL	

ADDRESSED COMMAND GROUP (ACG)

UNIVERSAL COMMAND GROUP (UCG)

LISTEN ADDRESS GROUP (LAG) (4)

TALK ADDRESS GROUP (TAG)

PRIMARY COMMAND GROUP (PCG)

SECONDARY COMMAND GROUP (SCG)

NOTES: (1) MSG = INTERFACE MESSAGE

(2) b_1 = DIO1...b_7 = DIO7

(3) REQUIRES SECONDARY COMMAND

(4) DENSE SUBSET (COLUMN 2 THROUGH 5)

Fig E1
Multiline Interface Messages: ISO-7 Bit Code Representation

Description des messages multilignes.

Table 37
Allowable Subsets to Controller Interface Function

Identification*	Capabilities											Notes	States Required										Other Requirements			Other Function Subsets Required				
	System Controller	Send IFC and Take Charge	Send REN	Respond to SRQ	Send I.F. Messages	Receive Control	Pass Control	Pass Control to Self	Parallel Poll	Take Control Synchronously			SNAS, SACS	SIIS, SIAS, SINS	SRIS, SRAS, SRNS	CSNS, CSRS	CACS, CSBS, CSHS, CSWS, CAWS	CADS	CIDS	CTRS	CPWS, CPPS	[TCT ∧ (ACDS) ∧ (TADS)]†	[(TADS)]‡	*tcs* not always false	C1	C2	AH1, L1-L4, or LE1-LE4	SH1	T1-T8, TE1-TE8
C0	N	N	N	N	N	N	N	N	N	N			O	O	O	O	O	O	O	O	O	O	O	O	O	–	–	–	–
C1	Y	–	–	–	–	–	–	–	–	–		(1)	R	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
C2	–	Y	–	–	–	–	–	–	–	–		(1),(6)	–	R	–	–	–	–	–	–	–	–	–	–	R	–	–	–	–	–
C3	–	–	Y	–	–	–	–	–	–	–		(1)	–	–	R	–	–	–	–	–	–	–	–	–	R	–	–	–	–	–
C4	–	–	–	Y	–	–	–	–	–	–		(1)	–	–	–	R	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
C5	–	–	–	–	Y	Y	Y	Y	Y	Y		(2),(3)	–	–	–	–	R	R	R	R	R	R	R	R	–	–	R	R	R
C6	–	–	–	–	Y	Y	Y	Y	Y	N		(2),(3)	–	–	–	–	R	R	R	R	R	R	R	O	–	–	–	R	R
C7	–	–	–	–	Y	Y	Y	Y	N	Y		(2),(3)	–	–	–	–	R	R	R	R	O	R	R	R	–	–	R	R	R
C8	–	–	–	–	Y	Y	Y	Y	N	N		(2),(3)	–	–	–	–	R	R	R	R	O	R	R	O	–	–	–	R	R
C9	–	–	–	–	Y	Y	Y	N	Y	Y		(2),(3)	–	–	–	–	R	R	R	R	R	R	O	R	–	–	R	R	R
C10	–	–	–	–	Y	Y	Y	N	Y	N		(2),(3)	–	–	–	–	R	R	R	R	R	R	O	O	–	–	–	R	R
C11	–	–	–	–	Y	Y	Y	N	N	Y		(2),(3)	–	–	–	–	R	R	R	R	O	R	O	R	–	–	R	R	R
C12	–	–	–	–	Y	Y	Y	N	N	N		(2),(3)	–	–	–	–	R	R	R	R	O	R	O	O	–	–	–	R	R
C13	–	–	–	–	Y	Y	N	N	Y	Y		(2)	–	–	–	–	R	R	R	O	R	O	O	R	–	R	R	R	–
C14	–	–	–	–	Y	Y	N	N	Y	N		(2)	–	–	–	–	R	R	R	O	R	O	O	O	–	R	–	R	–
C15	–	–	–	–	Y	Y	N	N	N	Y		(2)	–	–	–	–	R	R	R	O	O	O	O	R	–	R	R	R	–
C16	–	–	–	–	Y	Y	N	N	N	N		(2)	–	–	–	–	R	R	R	O	O	O	O	O	–	R	–	R	–
C17	–	–	–	–	Y	N	Y	Y	Y	Y		(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	R	R	–	–	R	R	R
C18	–	–	–	–	Y	N	Y	Y	Y	N		(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	R	O	–	–	–	R	R
C19	–	–	–	–	Y	N	Y	Y	N	Y		(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	R	R	–	–	R	R	R
C20	–	–	–	–	Y	N	Y	Y	N	N		(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	R	O	–	–	–	R	R
C21	–	–	–	–	Y	N	Y	N	Y	Y		(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	O	R	–	–	R	R	R
C22	–	–	–	–	Y	N	Y	N	Y	N		(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	O	O	–	–	–	R	R
C23	–	–	–	–	Y	N	Y	N	N	Y		(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	O	R	–	–	R	R	R
C24	–	–	–	–	Y	N	Y	N	N	N		(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	O	O	–	–	–	R	R
C25	–	–	–	–	Y	N	N	N	Y	Y		(2),(5)	–	–	–	–	R	O	O	O	R	O	O	R	–	–	R	R	–
C26	–	–	–	–	Y	N	N	N	Y	N		(2),(5)	–	–	–	–	R	O	O	O	R	O	O	O	–	–	–	R	–
C27	–	–	–	–	Y	N	N	N	N	Y		(2),(5)	–	–	–	–	R	O	O	O	O	O	O	R	–	–	R	R	–
C28	–	–	–	–	Y	N	N	N	N	N		(2),(5)	–	–	–	–	R	O	O	O	O	O	O	O	–	–	–	R	–

*Typical notation to describe a controller consists of the letter C followed by one or more of the numbers indicating the subsets selected. For example; C1, 2, 3, 4, 8.

†This is part of the CIDS to CADS transitional expression.

‡This is part of the CACS to CTRS transitional expression.

NOTES:

(1) One or more of subsets C1 through C4 may be chosen in any combination with any one of C5 through C28.

(2) Only one subset may be chosen from C5 through C28.

(3) The CTRS state shall be included in devices which are to be operated in multicontroller systems.

(4) These subsets are not allowed unless C2 is included.

(5) These subsets are intended to be used in devices and systems where no control passage is possible.

(6) When a system controller asserts IFC during the time another physical device is operating as controller-in-charge, the system controller should refrain from active assertion of the source handshake and ATN until the removal of the IFC message to preclude multiple controller contention.

O = omit, R = required, hyphen = not applicable or not required, Y = yes, N = no.

C1. General Comments

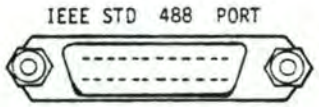
Sections 5 and 6 of this standard point out device designer responsibility to identify and device user responsibility to be familiar with the interface capabilities of each device containing IEEE Std 488-1978 interface functions. It is therefore recommended that each device be marked with an explicit code which, in effect, identifies those interface functions and subsets implemented within that device.

The reader is further reminded that full operational systems require detailed knowledge of the device dependent characteristics of each device in a system. These specifications are beyond the scope of this standard.

C2. Capability Identification Codes

It is recommended that an IEEE Std 488-1978 capability code be placed below or near the interface connector on each device to identify the complete set of interface functions contained within that device. In this standard each interface function and allowable subset thereof has an equivalent alphanumeric code to identify that particular capability. All such interface function capability codes may be expressed in a concise alphanumeric string and marked on the exterior of the device to facilitate user system assembly.

For example, a device with the basic talker function, the ability to send status bytes, the basic listener function, a listen only mode switch, service request capability, remote local capability without local lockout, manual configuration of the parallel poll capability, complete device clear capability, no capability for device trigger, and no controller capability would be identified with the following code:



SH1, AH1, T2, L1, SR1, RL2, PP2, DC1, DT0, C0, E1

The code identifies the eight specific interface functions implemented. In addition, the type of electrical interface contained within the device is specified. The notation E1 is used to identify open collector drivers and E2 is used to identify three-state drivers.

The device designer can place any further device capability information, useful for system configuration, at the appropriate places on the physical equipment and in the relevant documentation for that equipment.

C3. SH Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
SH0	no capability	all	none	none
SH1	complete capability	none	none	T1-T8, TE1-TE8, or C5-C28

C4. AH Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
AH0	no capability	all	none	none
AH1	complete capability	none	none	none

C5. T Function Allowable Subsets

Identification	Description				States Omitted	Other Requirements	Other Function Subsets Required
	Capabilities						
	Basic Talker	Serial Poll	Talk Only Mode	Unaddress If MLA			
T0	N	N	N	N	all	none	none
T1	Y	Y	Y	N	none	omit [MLA ∧ (ACDS)]	SH1 and AH1
T2	Y	Y	N	N	none	omit [MLA ∧ (ACDS)] ton always false	SH1 and AH1
T3	Y	N	Y	N	SPIS, SPMS, SPAS	omit [MLA ∧ (ACDS)]	SH1 and AH1
T4	Y	N	N	N	SPIS, SPMS, SPAS	omit [MLA ∧ (ACDS)] ton always false	SH1 and AH1
T5	Y	Y	Y	Y	none	include [MLA ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
T6	Y	Y	N	Y	none	include [MLA ∧ (ACDS)] ton always false	SH1 and L1-L4 or LE1-LE4
T7	Y	N	Y	Y	SPIS, SPMS, SPAS	include [MLA ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
T8	Y	N	N	Y	SPIS, SPMS, SPAS	include [MLA ∧ (ACDS)] ton always false	SH1 and L1-L4 or LE1-LE4

C6. T Function (with Address Extension) Allowable Subsets

Identification	Description				States Omitted	Other Requirements	Other Function Subsets Required
	Capabilities						
	Basic Extended Talker	Serial Poll	Talk Only Mode	Unaddress If MSA ∧ (LPAS)			
TE0	N	N	N	N	all	none	none
TE1	Y	Y	Y	N	none	omit [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and AH1
TE2	Y	Y	N	N	none	omit [MSA ∧ (LPAS) ∧ (ACDS)] ton always false	SH1 and AH1
TE3	Y	N	Y	N	SPIS, SPMS, SPAS	omit [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and AH1
TE4	Y	N	N	N	SPIS, SPMS, SPAS	omit [MSA ∧ (LPAS) ∧ (ACDS)] ton always false	SH1 and AH1
TE5	Y	Y	Y	Y	none	include [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
TE6	Y	Y	N	Y	none	include [MSA ∧ (LPAS) ∧ (ACDS)] ton always false	SH1 and L1-L4 or LE1-LE4
TE7	Y	N	Y	Y	SPIS, SPMS, SPAS	include [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
TE8	Y	N	N	Y	SPIS, SPMS, SPAS	include [MSA ∧ (LPAS) ∧ (ACDS)] ton always false	SH1 and L1-L4 or LE1-LE4

C7. L Function Allowable Subsets

Identification	Description			States Omitted	Other Requirements	Other Function Subsets Required
	Capabilities					
	Basic Listener	Listen Only Mode	Unaddress If MTA			
L0	N	N	N	all	none	none
L1	Y	Y	N	none	omit [MTA ∧ (ACDS)]	AH1
L2	Y	N	N	none	omit [MTA ∧ (ACDS)] lon always false	AH1
L3	Y	Y	Y	none	include [MTA ∧ (ACDS)]	AH1 and T1-T8 or TE1-TE8
L4	Y	N	Y	none	include [MTA ∧ (ACDS)] lon always false	AH1 and T1-T8 or TE1-TE8

C8. L Function (with Address Extension) Allowable Subsets

Identification	Description			States Omitted	Other Requirements	Other Function Subsets Required
	Capabilities					
	Basic Extended Listener	Listen Only Mode	Unaddress If MSA ∧ (TPAS)*			
LE0	N	N	N	all	none	none
LE1	Y	Y	N	none	omit [MSA ∧ (TPAS) ∧ (ACDS)]	AH1
LE2	Y	N	N	none	omit [MSA ∧ (TPAS) ∧ (ACDS)] lon always false	AH1
LE3	Y	Y	Y	none	include [MSA ∧ (TPAS)] ∧ (ACDS)]	AH1 and T1-T8 or TE1-TE8
LE4	Y	N	Y	none	include [MSA ∧ (TPAS) ∧ (ACDS)] lon always false	AH1 and T1-T8 or TE1-TE8

*Replaced by MTA when used together with the T function.

C9. SR Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
SR0	no capability	all	none	none
SR1	complete capability	none	none	T1, T2, T5, T6, TE1, TE2, TE5 or TE6

C10. RL Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
RL0	no capability	all	none	none
RL1	complete capability	none	none	L1-L4 or LE1-LE4
RL2	no local lock out	LWLS and RWLS	rtl always false	L1-L4 or LE1-LE4

C11. PP Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
PP0	no capability	all	none	none
PP1	remote configuration	none	include [((PPD ∧ (PACS) ∨ PPU) ∧ (ACDS)] include [PPE ∧ (PACS) ∧ (ACDS)] exclude lpe	L1-L4 or LE1-LE4
PP2	local configuration	PUCS, PACS	include lpe exclude [((PPD ∧ (PACS)) ∨ PPU) ∧ (ACDS)] exclude [PPE ∧ (PACS) ∧ (ACDS)] local messages shall be substituted for S, P1, P2, P3	none

C12. DC Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
DC0	no capability	all	none	none
DC1	complete capability	none	none	L1-L4 or LE1-LE4
DC2	omit selective device clear	none	omit [SDC ∧ (LADS)]	AH1

C13. DT Function Allowable Subsets

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
DT0	no capability	all	none	none
DT1	complete capability	none	none	L1-L4 or LE1-LE4

C14. C Function Allowable Subsets

Identification*	Capabilities										Notes	States Required									Other Requirements			Other Function Subsets Required			
	System Controller	Send IFC and Take Charge	Send REN	Respond to SRQ	Send I.F. Messages	Receive Control	Pass Control	Pass Control to Self	Parallel Poll	Take Control Synchronously		SNAS, SACS	SIIS, SIAS, SINS	SRIS, SRAS, SRNS	CSNS, CSRS	CACS, CSBS, CSHS, CSWS, CAWS	CADS	CIDS	CTRS	CPWS, CPPS	[TCT ∧ (ACDS) ∧ (TADS)]†	[(TADS)]‡	tcs not always false	C1	C2	AH1, L1-L4, or LE1-LE4	SH1	T1-T8, TE1-TE8
CØ	N	N	N	N	N	N	N	N	N	N		O	O	O	O	O	O	O	O	O	O	O	O	O	–	–	–	–
C1	Y	–	–	–	–	–	–	–	–	–	(1)	R	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
C2	–	Y	–	–	–	–	–	–	–	–	(1),(6)	–	R	–	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C3	–	–	Y	–	–	–	–	–	–	–	(1)	–	–	R	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C4	–	–	–	Y	–	–	–	–	–	–	(1)	–	–	–	R	–	–	–	–	–	–	–	–	–	–	–	–	–
C5	–	–	–	–	Y	Y	Y	Y	Y	Y	(2),(3)	–	–	–	–	R	R	R	R	R	R	R	R	–	–	R	R	R
C6	–	–	–	–	Y	Y	Y	Y	Y	N	(2),(3)	–	–	–	–	R	R	R	R	R	R	R	O	–	–	–	R	R
C7	–	–	–	–	Y	Y	Y	Y	N	Y	(2),(3)	–	–	–	–	R	R	R	R	O	R	R	R	–	–	R	R	R
C8	–	–	–	–	Y	Y	Y	Y	N	N	(2),(3)	–	–	–	–	R	R	R	R	O	R	R	O	–	–	–	R	R
C9	–	–	–	–	Y	Y	Y	N	Y	Y	(2),(3)	–	–	–	–	R	R	R	R	R	R	O	R	–	–	R	R	R
C10	–	–	–	–	Y	Y	Y	N	Y	N	(2),(3)	–	–	–	–	R	R	R	R	R	R	O	O	–	–	–	R	R
C11	–	–	–	–	Y	Y	Y	N	N	Y	(2),(3)	–	–	–	–	R	R	R	R	O	R	O	R	–	–	R	R	R
C12	–	–	–	–	Y	Y	Y	N	N	N	(2),(3)	–	–	–	–	R	R	R	R	O	R	O	O	–	–	–	R	R
C13	–	–	–	–	Y	Y	N	N	Y	Y	(2)	–	–	–	–	R	R	R	O	R	O	O	R	–	R	R	R	–
C14	–	–	–	–	Y	Y	N	N	Y	N	(2)	–	–	–	–	R	R	R	O	R	O	O	O	–	R	–	R	–
C15	–	–	–	–	Y	Y	N	N	N	Y	(2)	–	–	–	–	R	R	R	O	O	O	O	R	–	R	R	R	–
C16	–	–	–	–	Y	Y	N	N	N	N	(2)	–	–	–	–	R	R	R	O	O	O	O	O	–	R	–	R	–
C17	–	–	–	–	Y	N	Y	Y	Y	Y	(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	R	R	–	–	R	R	R
C18	–	–	–	–	Y	N	Y	Y	Y	N	(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	R	O	–	–	–	R	R
C19	–	–	–	–	Y	N	Y	Y	N	Y	(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	R	R	–	–	R	R	R
C20	–	–	–	–	Y	N	Y	Y	N	N	(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	R	O	–	–	–	R	R
C21	–	–	–	–	Y	N	Y	N	Y	Y	(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	O	R	–	–	R	R	R
C22	–	–	–	–	Y	N	Y	N	Y	N	(2),(3),(4)	–	–	–	–	R	O	R	R	R	R	O	O	–	–	–	R	R
C23	–	–	–	–	Y	N	Y	N	N	Y	(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	O	R	–	–	R	R	R
C24	–	–	–	–	Y	N	Y	N	N	N	(2),(3),(4)	–	–	–	–	R	O	R	R	O	R	O	O	–	–	–	R	R
C25	–	–	–	–	Y	N	N	N	Y	Y	(2),(5)	–	–	–	–	R	O	O	O	R	O	O	R	–	–	R	R	–
C26	–	–	–	–	Y	N	N	N	Y	N	(2),(5)	–	–	–	–	R	O	O	O	R	O	O	O	–	–	–	R	–
C27	–	–	–	–	Y	N	N	N	N	Y	(2),(5)	–	–	–	–	R	O	O	O	O	O	O	R	–	–	R	R	–
C28	–	–	–	–	Y	N	N	N	N	N	(2),(5)	–	–	–	–	R	O	O	O	O	O	O	O	–	–	–	R	–

*Typical notation to describe a controller consists of the letter C followed by one or more of the numbers indicating the subsets selected. For example; C1, 2, 3, 4, 8.

†This is part of the CIDS to CADS transitional expression.

‡This is part of the CACS to CTRS transitional expression.

NOTES:

(1) One or more of subsets C1 through C4 may be chosen in any combination with any one of C5 through C28.

(2) Only one subset may be chosen from C5 through C28.

(3) The CTRS state must be included in devices which are to be operated in multicontroller systems.

(4) These subsets are not allowed unless C2 is included.

(5) These subsets are intended to be used in devices and systems where no control passage is possible.

(6) When a system controller asserts IFC during the time another physical device is operating as controller-in-charge, the system controller should refrain from active assertion of the source handshake and ATN until the removal of the IFC message to preclude multiple controller contention.

O = omit, R = required, hyphen = not applicable or not required, Y = yes, N = no.

Répertoire des messages d' interfacage

Mnemonic	Message	Interface Function(s)
LOCAL MESSAGES RECEIVED (by interface functions)		
gts	go to standby	C
ist	individual status qualifier	PP
lon	listen only	L, LE
[lpe]	local poll enable	PP
ltn	listen	L, LE
lun	local unlisten	L, LE
nba	new byte available	SH
pon	power on	SH, AH, T, TE, L, LE, SR, RL, PP, C
rdy	ready	AH
rpp	request parallel poll	C
rsc	request system control	C
rsv	request service	SR
rtl	return to local	RL
sic	send interface clear	C
sre	send remote enable	C
tca	take control asynchronously	C
tcs	take control synchronously	AH, C
ton	talk only	T, TE
LOCAL MESSAGES SENT (to device functions)		
	None defined; see Message Output tables in Section 2 for description of Device Function Interaction which provides guidelines as to the appropriate states from which local messages may be sent to the device functions.	
REMOTE MESSAGES RECEIVED		
ATN	attention	SH, AH, T, TE, L, LE, PP, C
DAB	data byte	(via L, LE)
DAC	data accepted	SH
DAV	data valid	AH
DCL	device clear	DC
END	end	(via L, LE)
GET	group execute trigger	DT
GTL	go to local	RL
IDY	identify	L, LE, PP
IFC	interface clear	T, TE, L, LE, C
LLO	local lockout	RL
MLA	my listen address	L, LE, RL
[MLA]	my listen address	T
MSA or [MSA]	my secondary address	TE, LE
MTA	my talk address	T, TE
[MTA]	my talk address	L
OSA	other secondary address	TE
OTA	other talk address	T, TE
PCG	primary command group	TE, LE, PP
PPC	parallel poll configure	PP
[PPD]	parallel poll disable	PP
[PPE]	parallel poll enable	PP
PPRn	parallel poll response n	(via C)
PPU	parallel poll unconfigure	PP
REN	remote enable	RL
RFD	ready for data	SH
RQS	request service	(via L, LE)

Interface Message Reference List (Continued)

Mnemonic	Message	Interface Function(s)
[SDC]	selected device clear	DC
SPD	serial poll disable	T, TE
SPE	serial poll enable	T, TE
SRQ	service request	(via C)
STB	status byte	(via L, LE)
TCT or [TCT]	take control	C
UNL	unlisten	L, LE
REMOTE MESSAGES SENT		
ATN	attention	C
DAB	data byte	(via T, TE)
DAC	data accepted	AH
DAV	data valid	SH
DCL	device clear	(via C)
END	end	(via T)
GET	group execute trigger	(via C)
GTL	go to local	(via C)
IDY	identify	C
IFC	interface clear	C
LLO	local lockout	(via C)
MLA or [MLA]	my listen address	(via C)
MSA or [MSA]	my secondary address	(via C)
MTA or [MTA]	my talk address	(via C)
OSA	other secondary address	(via C)
OTA	other talk address	(via C)
PCG	primary command group	(via C)
PPC	parallel poll configure	(via C)
[PPD]	parallel poll disable	(via C)
[PPE]	parallel poll enable	(via C)
PPRn	parallel poll response n	PP
PPU	parallel poll unconfigure	(via C)
REN	remote enable	C
RFD	ready for data	AH
RQS	request service	T, TE
[SDC]	selected device clear	(via C)
SPD	serial poll disable	(via C)
SPE	serial poll enable	(via C)
SRQ	service request	SR
STB	status byte	(via T, TE)
TCT	take control	(via C)
UNL	unlisten	(via C)
UNT	untalk	(via C)

Conventions de codification des messages distants

Mnemonic	Message Name		Type	Class	Bus Signal Line(s) and Coding That Asserts the True Value of the Message: DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	NDAC	NRFD	ATN	EOI	SRQ	IFC	REN
ACG	addressed command group		M	AC	Y	Ø	Ø	Ø	X	X	X	X	XXX	1	X	X	X	X	X
ATN	attention		U	UC	X	X	X	X	X	X	X	X	XXX	1	X	X	X	X	X
DAB	data byte	(Notes 1, 9)	M	DD	D8	D7	D6	D5	D4	D3	D2	D1	XXX	Ø	X	X	X	X	X
DAC	data accepted		U	HS	X	X	X	X	X	X	X	X	XXØ	X	X	X	X	X	X
DAV	data valid		U	HS	X	X	X	X	X	X	X	X	1XX	X	X	X	X	X	X
DCL	device clear		M	UC	Y	Ø	Ø	1	Ø	1	Ø	Ø	XXX	1	X	X	X	X	X
END	end	(Note 9)	U	ST	X	X	X	X	X	X	X	X	XXX	Ø	1	X	X	X	X
EOS	end of string	(Notes 2, 9)	M	DD	E8	E7	E6	E5	E4	E3	E2	E1	XXX	Ø	X	X	X	X	X
GET	group execute trigger		M	AC	Y	Ø	Ø	Ø	1	Ø	Ø	Ø	XXX	1	X	X	X	X	X
GTL	go to local		M	AC	Y	Ø	Ø	Ø	Ø	Ø	Ø	1	XXX	1	X	X	X	X	X
IDY	identify		U	UC	X	X	X	X	X	X	X	X	XXX	X	1	X	X	X	X
IFC	interface clear		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	1	X	X
LAG	listen address group		M	AD	Y	Ø	1	X	X	X	X	X	XXX	1	X	X	X	X	X
LLO	local lock out		M	UC	Y	Ø	Ø	1	Ø	Ø	Ø	1	XXX	1	X	X	X	X	X
MLA	my listen address	(Note 3)	M	AD	Y	Ø	1	L5	L4	L3	L2	L1	XXX	1	X	X	X	X	X
MTA	my talk address	(Note 4)	M	AD	Y	1	Ø	T5	T4	T3	T2	T1	XXX	1	X	X	X	X	X
MSA	my secondary address	(Note 5)	M	SE	Y	1	1	S5	S4	S3	S2	S1	XXX	1	X	X	X	X	X
NUL	null byte		M	DD	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	XXX	X	X	X	X	X	X
OSA	other secondary address		M	SE	(OSA = SCG ∧ $\overline{\text{MSA}}$)														
OTA	other talk address		M	AD	(OTA = TAG ∧ $\overline{\text{MTA}}$)														
PCG	primary command group		M	—	(PCG = ACG ∨ UCG ∨ LAG ∨ TAG)														
PPC	parallel poll configure		M	AC	Y	Ø	Ø	Ø	Ø	1	Ø	1	XXX	1	X	X	X	X	X
PPE	parallel poll enable	(Note 6)	M	SE	Y	1	1	Ø	S	P3	P2	P1	XXX	1	X	X	X	X	X
PPD	parallel poll disable	(Note 7)	M	SE	Y	1	1	1	D4	D3	D2	D1	XXX	1	X	X	X	X	X
PPR1	parallel poll response 1	(Note 10)	U	ST	X	X	X	X	X	X	X	1	XXX	1	1	X	X	X	X
PPR2	parallel poll response 2	(Note 10)	U	ST	X	X	X	X	X	X	1	X	XXX	1	1	X	X	X	X
PPR3	parallel poll response 3	(Note 10)	U	ST	X	X	X	X	X	1	X	X	XXX	1	1	X	X	X	X
PPR4	parallel poll response 4	(Note 10)	U	ST	X	X	X	X	1	X	X	X	XXX	1	1	X	X	X	X
PPR5	parallel poll response 5	(Note 10)	U	ST	X	X	X	1	X	X	X	X	XXX	1	1	X	X	X	X

Mnemonic	Message Name		Type	Class	Bus Signal Line(s) and Coding That Asserts the True Value of the Message: DIO8	7	6	5	4	3	2	1	NDAC	NRFD	ATN	EOI	SRQ	IFC	REN
PPR6	parallel poll response 6	(Note 10)	U	ST	X	X	1	X	X	X	X	X	XXX	1	1	X	X	X	
PPR7	parallel poll response 7	(Note 10)	U	ST	X	1	X	X	X	X	X	X	XXX	1	1	X	X	X	
PPR8	parallel poll response 8	(Note 10)	U	ST	1	X	X	X	X	X	X	X	XXX	1	1	X	X	X	
PPU	parallel poll unconfigure		M	UC	Y	Ø	Ø	1	Ø	1	Ø	1	XXX	1	X	X	X	X	
REN	remote enable		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	X	1	
RFD	ready for data		U	HS	X	X	X	X	X	X	X	X	XØX	X	X	X	X	X	
RQS	request service	(Note 9)	U	ST	X	1	X	X	X	X	X	X	XXX	Ø	X	X	X	X	
SCG	secondary command group		M	SE	Y	1	1	X	X	X	X	X	XXX	1	X	X	X	X	
SDC	selected device clear		M	AC	Y	Ø	Ø	Ø	Ø	1	Ø	Ø	XXX	1	X	X	X	X	
SPD	serial poll disable		M	UC	Y	Ø	Ø	1	1	Ø	Ø	1	XXX	1	X	X	X	X	
SPE	serial poll enable		M	UC	Y	Ø	Ø	1	1	Ø	Ø	Ø	XXX	1	X	X	X	X	
SRQ	service request		U	ST	X	X	X	X	X	X	X	X	XXX	X	X	1	X	X	
STB	status byte	(Notes 8, 9)	M	ST	S8	X	S6	S5	S4	S3	S2	S1	XXX	Ø	X	X	X	X	
TCT	take control		M	AC	Y	Ø	Ø	Ø	1	Ø	Ø	1	XXX	1	X	X	X	X	
TAG	talk address group		M	AD	Y	1	Ø	X	X	X	X	X	XXX	1	X	X	X	X	
UCG	universal command group		M	UC	Y	Ø	Ø	1	X	X	X	X	XXX	1	X	X	X	X	
UNL	unlisten		M	AD	Y	Ø	1	1	1	1	1	1	XXX	1	X	X	X	X	
UNT	untalk	(Note 11)	M	AD	Y	1	Ø	1	1	1	1	1	XXX	1	X	X	X	X	

The 1/Ø coding on ATN when sent concurrent with multiline messages has been added to this revision for interpretive convenience.

NOTES:

(1) D1-D8 specify the device dependent data bits.

(2) E1-E8 specify the device dependent code used to indicate the EOS message.

(3) L1-L5 specify the device dependent bits of the device's listen address.

(4) T1-T5 specify the device dependent bits of the device's talk address.

(5) S1-S5 specify the device dependent bits of the device's secondary address.

(6) S specifies the sense of the PPR.

S	Response
Ø	Ø
1	1

P1-P3 specify the PPR message to be sent when a parallel poll is executed.

P3	P2	P1	PPR Message
Ø	Ø	Ø	PPR1
.	.	.	.
.	.	.	.
.	.	.	.
1	1	1	PPR8

(7) D1-D4 specify don't-care bits that shall not be decoded by the receiving device. It is recommended that all zeroes be sent.

(8) S1-S6, S8 specify the device dependent status. (DIO7 is used for the RQS message.)

(9) The source of the message on the ATN line is always the C function, whereas the messages on the DIO and EOI lines are enabled by the T function.

(10) The source of the messages on the ATN and EOI lines is always the C function, whereas the source of the messages on the DIO lines is always the PP function.

(11) This code is provided for system use, see 6.3.

Examples de formats de messages issus des Recommandations I.E.E.E.-728.

(These Appendixes are not a part of IEEE Std 728-1982, IEEE Recommended Practice for Code and Format Conventions for Use with ANSI/IEEE Std 488-1978, IEEE Standard Digital Interface for Programmable Instrumentations.)

Appendix A
Message Format Examples: Syntax Diagram Uses

A1. General Comments

The conventions presented throughout this recommended practice provide the designer and user with a set of recommended formats for each different message type.

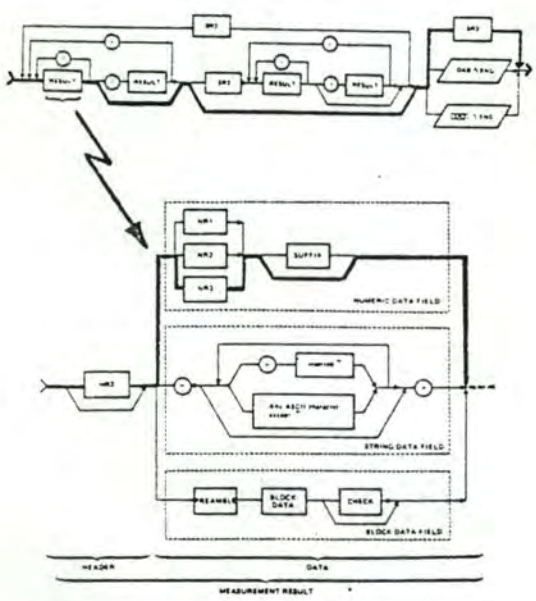
Rigid and mandatory message structures are considered to be too restrictive for broad general use. Typical formats for several different messages are given in this Appendix to illustrate the application of the recommended conventions. It is expected that individual product designs may deviate from the recommended conventions in special circumstances. Adherence to the recommended formats if encouraged strongly to increase the level of information interchange and compatibility among interconnected devices.

A1.1 Measurement Messages. The Figures in this section are in part similar to Figs 21(a) and 21(b).

A1.1.1 One voltage measurement of +10.002 VDC is taken on the 10 V range and expressed in NR3 (exponential) notation. The numeric value is preceded by an alpha header to indicate the type (DC) measured result. The message is concluded with CRLF for one measurement result.

Measurement Result:
DC+10002.E-03 CRLF
Syntax Construction:

Fig A1



Pass 1
Separator CRLF

Pass 1
Header DC
Data+10002.E-03

Fig A2

A1.1.2 A spectrum analysis is made over a specified frequency span yielding 1000 measurements expressed in NR2 notation and separated by commas (in this case the dBm units are implicit).

Measurement Results:

−10.5,−11.1,(...)−86.1,(...)−11.1 [1 Λ END]

NOTE: The (...) symbols do not appear in the actual message. They serve only to indicate additional data in this example. [ ] indicates END sent concurrent with last byte [1] of *result*.

Syntax Construction:

Fig A1

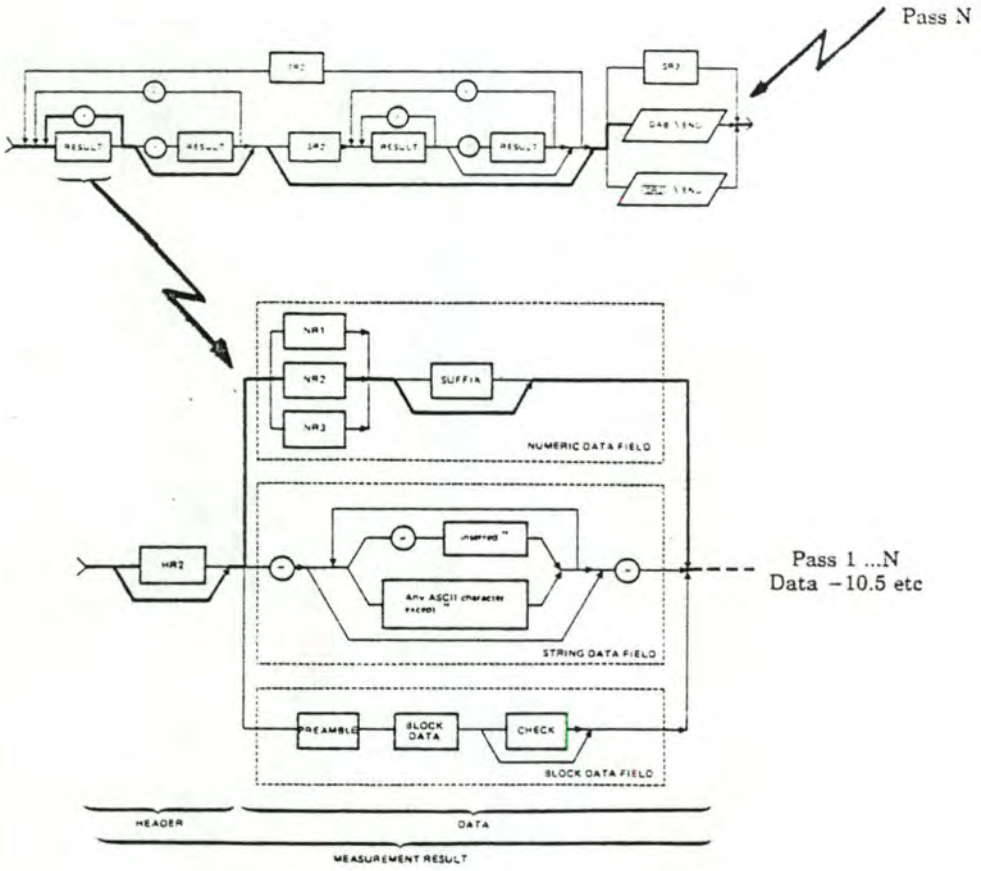


Fig A2

A1.1.3 A frequency counter with two channels (A and B) measures frequencies of 4.23 kHz and 2.60 kHz.

Measurement Result:

AFKHZ4.23,BFKHZ2.60 NL

Syntax Construction:

Fig A1

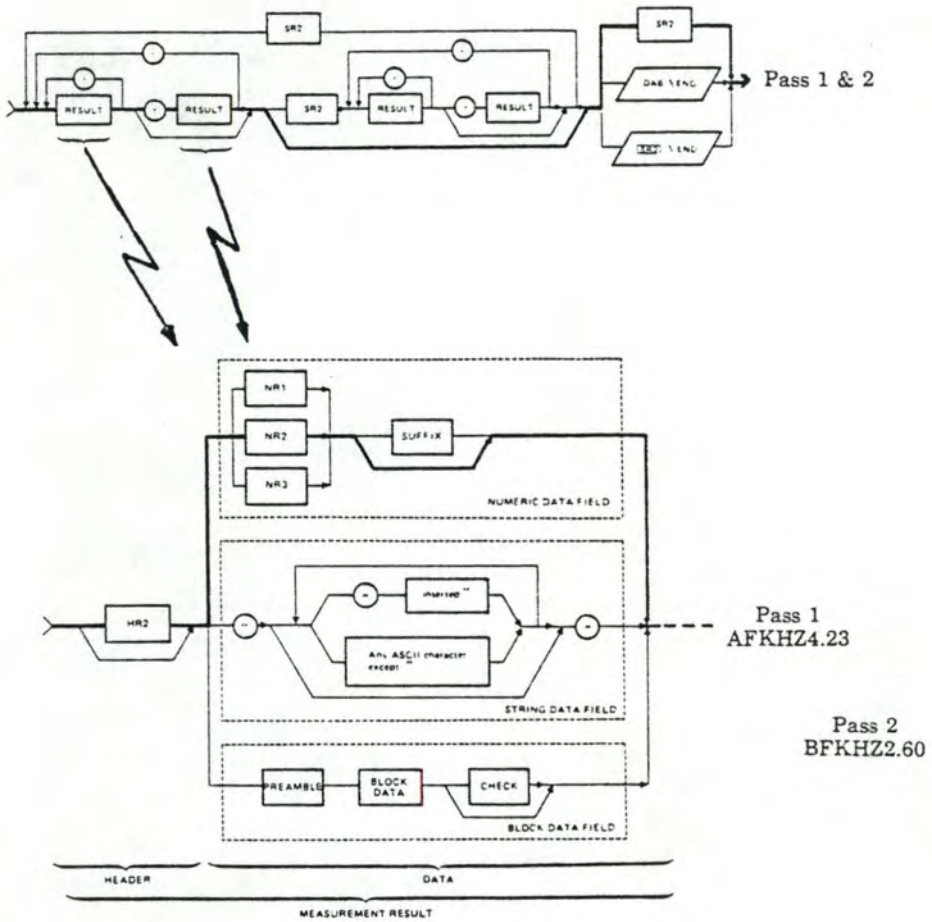


Fig A2

A1.1.4 A DVM takes a measurement of 10.002 V dc and sends it in NR2 notation.

Measurement result:

VOLTS 10.00[2 ΛEND]

Syntax Construction:

Fig A1

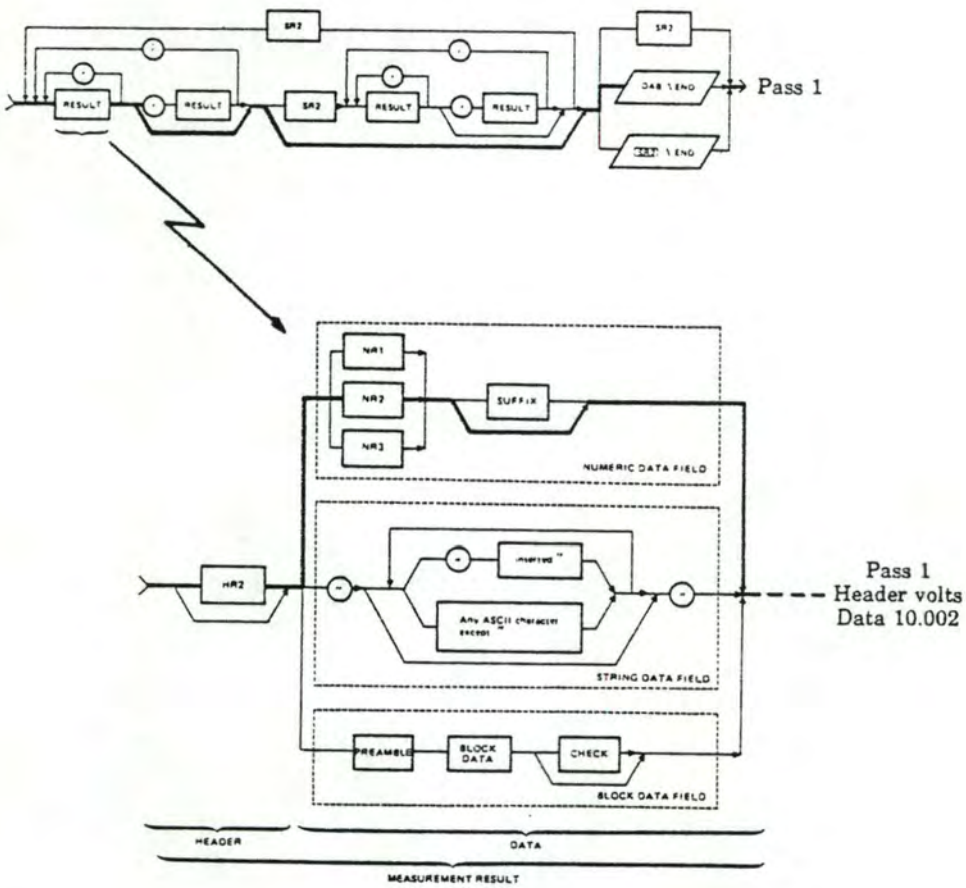


Fig A2

A1.1.5 An oscilloscope sends a waveform identification as a string value and the waveform itself (1024 data points) as a block value (requires two passes through the syntax diagram separated by ! , !).

Measurement Results:

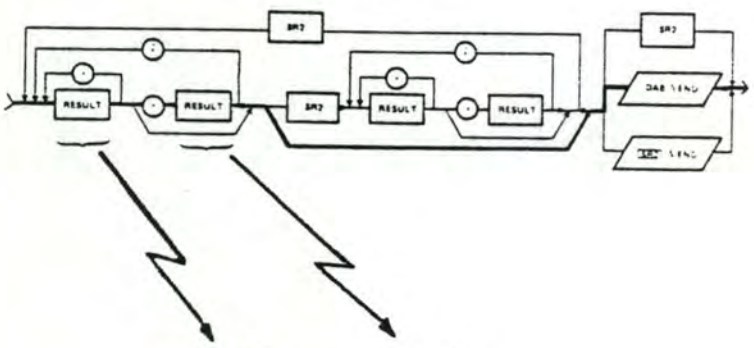
WAVEFORM "CHANNEL 1",
#CLLD . . . DDD(CRC1)(CRC2)

where:

L = 0000 0100 (length byte 1)
L =0000 0010 (length byte 2) (total length 1026 bytes)
D =0110 0000 (point 1)
: = :
D =0110 1000 (point 1022)
D =1011 0000 (point 1023)
D =1000 1101 (point 1024)
CRC1 =xxxx xxxx (CRC check byte 1)
CRC2 =xxxx xxxx (CRC check byte 2)

Syntax Construction:

Fig A1



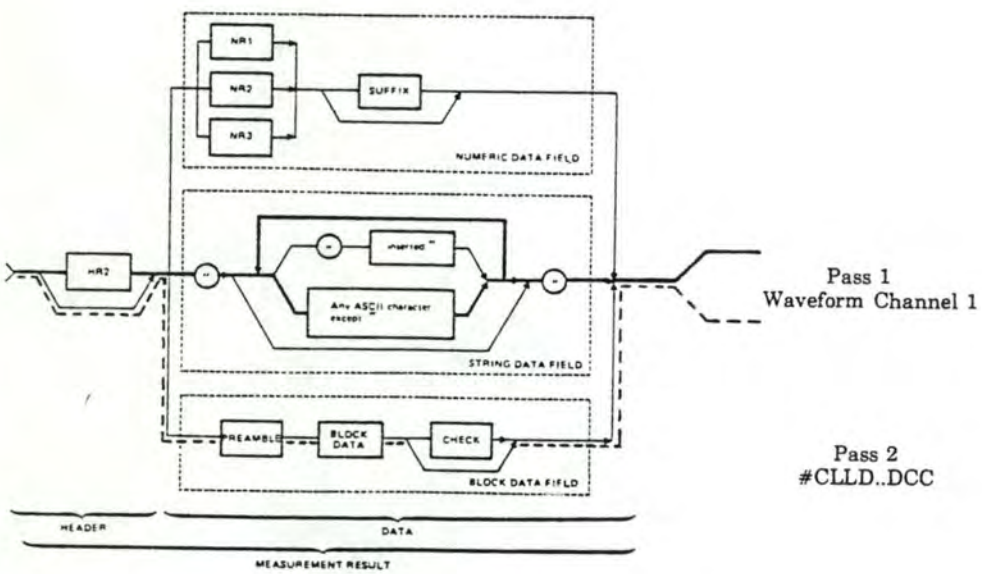


Fig A2