

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

60823

Première édition
First edition
1990-10

**Bus système à microprocesseurs (VMSbus) –
Bus sous-système série du Bus CEI 821
(VMEbus)**

**Microprocessor system bus (VMSbus) –
Serial sub-system bus of the IEC 821 Bus
(VMEbus)**



Numéro de référence
Reference number
CEI/IEC 60823: 1990

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles dans le Catalogue de la CEI.

Les renseignements relatifs à des questions à l'étude et des travaux en cours entrepris par le comité technique qui a établi cette publication, ainsi que la liste des publications établies, se trouvent dans les documents ci-dessous:

- «Site web» de la CEI*
- **Catalogue des publications de la CEI**
Publié annuellement et mis à jour régulièrement
(Catalogue en ligne)*
- **Bulletin de la CEI**
Disponible à la fois au «site web» de la CEI et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International (VEI)*.

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique*, la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles*, et la CEI 60617: *Symboles graphiques pour schémas*.

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
Published yearly with regular updates
(On-line catalogue)*
- **IEC Bulletin**
Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary (IEV)*.

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams*.

* See web site address on title page.

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

60823

Première édition
First edition
1990-10

**Bus système à microprocesseurs (VMSbus) –
Bus sous-système série du Bus CEI 621
(VMEbus)**

**Microprocessor system bus (VMSbus) –
Serial sub-system bus of the IEC 821 bus
(VMEbus)**

© IEC 1990 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembé Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE **XG**

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

	Pages
PREAMBULE	14
PREFACE	14

CHAPITRE 0: DOMAINE D'APPLICATION

CHAPITRE 1: INTRODUCTION A LA NORME CEI 823 VMSbus

Sections

1.1 Objectifs du bus série	18
1.1.1 Objectifs de la norme CEI 823 VMSbus	18
1.2 Eléments du système constituant l'interface du bus série	20
1.3 Terminologie relative à la norme du bus série	24
1.4 Organisation de la norme	26
1.5 Relations entre le bus série et un bus parallèle du fond de panier	26

CHAPITRE 2: GENERALITES SUR LE BUS SERIE

2.1 Généralités sur la couche physique	34
2.1.1 Signaux et modules de la couche physique	34
2.1.2 Utilisation des signaux de la couche physique	34
2.2 Généralités sur la couche de liaison de données	38
2.2.1 Modules et interfaces	38
2.2.2 Groupes de modules	42
2.2.3 Protocole de transmission d'une trame	44
2.3 Utilisation du bus série pour transférer des données	46
2.4 Utilisation du bus série pour positionner à un et à zéro des indicateurs	46
2.5 Applications des groupes de modules du bus série	46
2.6 Possibilités d'adressage du bus série	48

CHAPITRE 3: TRAMES ET SOUS-TRAMES DU BUS SERIE

3.1 Types généraux de trames	50
3.2 En-tête	56
3.3 Sous-trame de type de trame	60
3.4 Sous-trame de données	64
3.5 Sous-trame d'état de trame	64
3.6 Sous-trame de détection de désynchronisation	68
3.7 Protocole de resynchronisation	68
3.8 Resynchronisation lors de l'initialisation	72

CONTENTS

	Page
FOREWORD	15
PREFACE	15

CHAPTER 0: SCOPE

CHAPTER 1: INTRODUCTION TO THE IEC 823 VMSbus Standard

Section

1.1 Serial bus objectives	19
1.1.1 IEC 823 VMSbus standard objectives	19
1.2 Serial bus interface system elements	21
1.3 Serial bus standard terminology	25
1.4 Standard organization	27
1.5 Standard relationship of the serial bus and a parallel back-plane bus	27

CHAPTER 2: SERIAL BUS OVERVIEW

2.1 Overview of the Physical Layer	35
2.1.1 Physical Layer modules and signals	35
2.1.2 Physical Layer signalling	35
2.2 Link Layer overview	39
2.2.1 Modules and interfaces	39
2.2.2 Module groups	43
2.2.3 Frame transmission protocol	45
2.3 Using the serial bus to transfer data	47
2.4 Using the serial bus to set and reset flags	47
2.5 Applications of serial bus module groups	47
2.6 Serial bus addressing capabilities	49

CHAPTER 3: SERIAL BUS FRAMES AND SUBFRAMES

3.1 Basic frame types	51
3.2 The Header	57
3.3 The Frame Type subframe	61
3.4 The Data subframe	65
3.5 The Frame Status subframe	65
3.6 The Jam Detect subframe	69
3.7 Jam protocol	69
3.8 Jam on Reset	73

Sections

Pages

CHAPITRE 4: COUCHE PHYSIQUE

4.1	Module GENERATEUR D'HORLOGE	74
4.1.1	Interface au support physique	74
4.1.2	Interface avec la couche physique et la couche de liaison	78
4.1.3	Fonctionnement	78
4.2	Module d'EXTENSION	78
4.2.1	Interfaces avec les supports physiques	78
4.2.2	Interfaces avec la couche physique et la couche de liaison	82
4.2.3	Fonctionnement	84
4.3	Module d'ACCES AU BUS	86
4.3.1	Interface avec le support physique	86
4.3.2	Interface avec la couche physique	86
4.3.3	Interface de la couche de liaison	92
4.3.4	Fonctionnement	92
4.4	Spécifications électriques	96
4.4.1	Caractéristiques d'entrée du support physique de fond de panier	96
4.4.2	Commande et charge pour SERCLK	98
4.4.3	Commande et charge pour SERDAT*	98
4.4.4	Caractéristiques d'entrée pour le support physique étendu	100
4.4.5	Commande du support physique étendu	100
4.4.6	Charge du support physique étendu	104
4.4.7	Résistance aux contraintes/détériorations du support physique étendu	106

CHAPITRE 5: MODULES DE LA COUCHE DE LIAISON DE DONNEES

5.1	Notations pour les diagrammes d'états	110
5.2	Module EMETTEUR D'EN-TETE	110
5.2.1	interface avec la couche physique	112
5.2.2	Interface avec la couche de liaison de données	112
5.2.3	Interface avec les couches de niveau supérieur	114
5.2.4	Initialisation	116
5.2.5	Fonctionnement	116
5.3	Module RECEPTEUR D'EN-TETE	122
5.3.1	interface avec la couche physique	124
5.3.2	Interface avec la couche de liaison de données	124
5.3.3	Interface avec les couches supérieures	124
5.3.4	Initialisation	128
5.3.5	Programmation de l'adresse	128
5.3.6	Fonctionnement	128

CHAPTER 4: THE PHYSICAL LAYER

4.1	CLOCK SOURCE module	75
4.1.1	Medium interface	75
4.1.2	Physical and Link Layer interface	79
4.1.3	Operation	79
4.2	BRIDGE module	79
4.2.1	Media interfaces	79
4.2.2	Physical and Link Layer interfaces	83
4.2.3	Operation	85
4.3	BUS ACCESS module	87
4.3.1	Medium interface	87
4.3.2	Physical Layer interface	87
4.3.3	Link Layer interface	93
4.3.4	Operation	93
4.4	Electrical specifications	97
4.4.1	Input characteristics for the Backplane Medium	97
4.4.2	Driving and loading for SERCLK	99
4.4.3	Driving and loading for SERDAT*	99
4.4.4	Input characteristics for the Extended Medium	101
4.4.5	Driving the Extended Medium	101
4.4.6	Loading on the Extended Medium	105
4.4.7	Stress/damage resistance for the Extended Medium	107

CHAPTER 5: LINK LAYER MODULES

5.1	State diagram notation	111
5.2	HEADER SENDER module	111
5.2.1	Physical Layer interface	113
5.2.2	Link Layer interface	113
5.2.3	Higher Layer interface	115
5.2.4	Initialization	117
5.2.5	Operation	117
5.3	HEADER RECEIVER module	123
5.3.1	Physical Layer interface	125
5.3.2	Link Layer interface	125
5.3.3	Higher Layer interface	125
5.3.4	Initialization	129
5.3.5	Programming the address	129
5.3.6	Operation	129

Sections	Pages
5.4 Module EMETTEUR DE DONNEES	136
5.4.1 Interface avec la couche physique	136
5.4.2 Interface avec la couche de liaison de données	136
5.4.3 Interface avec la couche supérieure	136
5.4.4 Initialisation	140
5.4.5 Programmation du port de données	140
5.4.6 Fonctionnement	143
5.5 Module RECEPTEUR DE DONNEES	150
5.5.1 Interface avec la couche physique	150
5.5.2 Interface avec la couche de liaison de données	150
5.5.3 Interface avec les couches de niveau supérieur	150
5.5.4 Initialisation	154
5.5.5 Lecture des données du port de données	154
5.5.6 Fonctionnement	156
5.6 GESTIONNAIRE DE TRAME	162
5.6.1 Interface avec la couche physique	162
5.6.2 Interface avec la couche de liaison de données	162
5.6.3 Interface avec les couches de niveau supérieur	164
5.6.4 Initialisation	166
5.6.5 Fonctionnement	168
CHAPITRE 6: GROUPES DE LA COUCHE DE LIAISON DE DONNÉES ET PROTOCOLE	
6.1 Groupes simples	174
6.1.1 Indicateur simple	174
6.1.2 Gestionnaire de signal virtuel	178
6.1.3 Parleur sur demande	182
6.1.4 Parleur transactionnel	187
6.1.5 Ecouteur sur demande	192
6.1.6 Ecouteur transactionnel	196
6.1.7 Indicateur multiadresse	200
6.1.8 Parleur multiadresse	202
6.1.9 Ecouteur multiadresse	206
6.1.10 Gestionnaire de priorité variable	210
6.2 Groupes composés	214
6.2.1 Gestionnaire d'écriture	214
6.2.2 Gestionnaire de lecture	218
6.2.3 Emetteur-récepteur de bus virtuel	222
6.2.4 Sémaphore	228
6.2.5 Sémaphore à vérification de signature	240
6.2.6 Groupe à passage de jeton	246
6.2.7 Gestionnaire d'écriture avec accusé de réception	252
6.2.8 Ecouteur transactionnel avec accusé de réception	256
6.2.9 Ecouteur transactionnel à verrouillage	260
6.2.10 Parleur transactionnel à verrouillage	266

Section	Page
5.4 DATA SENDER module	137
5.4.1 Physical Layer interface	137
5.4.2 Link Layer interface	137
5.4.3 Higher Layer interface	137
5.4.4 Initialization	141
5.4.5 Programming the Data Port	141
5.4.6 Operation	143
5.5 DATA RECEIVER module	151
5.5.1 Physical Layer interface	151
5.5.2 Link Layer interface	151
5.5.3 Higher Layer interface	151
5.5.4 Initialization	155
5.5.5 Reading data from the Data Port	155
5.5.6 Operation	157
5.6 FRAME MONITOR module	163
5.6.1 Physical Layer interface	163
5.6.2 Link Layer interface	163
5.6.3 Higher Layer interface	165
5.6.4 Initialization	167
5.6.5 Operation	169

CHAPTER 6: LINK LAYER GROUPS AND PROTOCOL

6.1 Simple groups	175
6.1.1 Simple Flag	175
6.1.2 Virtual Signal Controller	179
6.1.3 On-Demand Talker	183
6.1.4 Transaction Talker	187
6.1.5 On-Demand Listener	193
6.1.6 Transaction Listener	197
6.1.7 Multiaddress Flag	201
6.1.8 Multiaddress Talker	203
6.1.9 Multiaddress Listener	207
6.1.10 Variable Priority Controller	211
6.2 Compound groups	215
6.2.1 Writing Controller	215
6.2.2 Reading Controller	219
6.2.3 Virtual Bus Transceiver	223
6.2.4 Semaphore	229
6.2.5 Signature-Checking Semaphore	241
6.2.6 Token Passing Group	247
6.2.7 Handshaking Writing Controller	253
6.2.8 Handshaking Transaction Listener	257
6.2.9 Locking Transaction Listener	261
6.2.10 Locking Transaction Talker	267

Sections

Pages

CHAPITRE 7: SUPPORT PHYSIQUE DE FOND DE PANIER
DU BUS CEI 821 VMEbus

7.1	Spécifications électriques supplémentaires	272
7.1.1	Résistances d'adaptation	272
7.2	Spécifications mécaniques	272
7.3	Paramètres de chronologie	272
7.3.1	Tests de conformité	278

CHAPITRE 8: SUPPORT PHYSIQUE D'EXTENSION

8.1	Spécifications électriques supplémentaires	282
8.1.1	Réseaux d'adaptation	282
8.1.2	La ligne BALANCE	282
8.1.3	Paramètre du câble	286
8.1.4	Filtrage et couplage optoélectrique	286
8.2	Spécifications mécaniques	286
8.2.1	Connecteurs et affectations des signaux	288
8.2.1.1	Connecteur D-sub à 9 broches	288
8.2.1.2	Connecteur pour câble plat à 10 broches et connecteur P2/J2	290
8.2.1.3	Mélanges des trois types de connecteurs	294
8.2.2	Longueur des segments de câble	306
8.3	Chronologie du support physique d'extension	306
8.3.1	Taux de transmission en fonction de la longueur	306
8.3.2	Paramètres de chronologie	308
8.3.3	Terminologie pour les signaux du support physique d'extension	316
8.3.4	Tests de conformité	318

Figures

2-1	Présentation en couches et découpage du système de bus série.	30
2-2	Présentation et interfaçage sur une carte typique	32
2-3	Configuration du support physique de fond de panier pour un châssis simple	36
2-4	Carte indépendante sur le support physique étendu	36
3-1	Types généraux de trame	52
4-1	Module GENERATEUR D'HORLOGE	76
4-2	Formes des signaux du bus d'extension	76
4-3	Signaux utilisés par le module d'EXTENSION	80
4-4	Formes des signaux du bus série	80
4-5	Signaux utilisés par le module d'ACCES AU BUS	90

Section

Page

CHAPTER 7: IEC 821 VMEbus BACKPLANE MEDIUM

7.1	Additional electrical specifications	273
7.1.1	Terminating resistors	273
7.2	Mechanical specifications	273
7.3	Timing parameters	273
7.3.1	Testing of compliance	279

CHAPTER 8: EXTENDED MEDIUM

8.1	Additional electrical specifications	283
8.1.1	Termination networks	283
8.1.2	The BALANCE line	283
8.1.3	Cable parameters	287
8.1.4	Filtering and optocoupling	287
8.2	Mechanical specifications	287
8.2.1	Connectors and signal assignments	289
8.2.1.1	9-pole D-sub connector	289
8.2.1.2	10-pole flat cable connector and P2/J2 connector	291
8.2.1.3	Mixtures of all three connector types	295
8.2.2	Cable segment length	307
8.3	Extended Medium timing	307
8.3.1	Data rate versus length	307
8.3.2	Timing parameters	309
8.3.3	Terminology for Extended Medium signals	317
8.3.4	Testing of compliance	319

Figures

2-1	Serial bus system structure and layering	31
2-2	Layering and interfacing on a typical board	33
2-3	Single-subrack Backplane Medium configuration	37
2-4	Free-standing board on the Extended Medium	37
3-1	Basic frame types	53
4-1	CLOCK SOURCE module	77
4-2	Extended bus waveforms	77
4-3	Signals used by the BRIDGE module	81
4-4	Serial bus waveforms	81
4-5	Signals used by the BUS ACCESS module	91

Figures

Pages

4-6	Formes des signaux du support physique de fond de panier ..	90
4-7	Circuit de charge de test pour la REGLE 4.26 et la REGLE 4.27	100
4-8	Circuit de charge de test pour la REGLE 4.28 et la REGLE 4.29	102
5-1	EMETTEUR D'EN-TETE avec GESTIONNAIRE DE TRAME	112
5-2	Diagramme d'état de l'EMETTEUR D'EN-TETE	120
5-3	Signaux utilisés par un RECEPTEUR D'EN-TETE	126
5-4	Diagramme d'état du RECEPTEUR D'EN-TETE	130
5-5	EMETTEUR DE DONNEES avec RECEPTEUR D'EN-TETE	138
5-6	Diagramme d'état de l'EMETTEUR DE DONNEES	146
5-7	RECEPTEUR DE DONNEES avec RECEPTEUR D'EN-TETE	152
5-8	Diagramme d'état du RECEPTEUR DE DONNEES	158
5-9	Diagramme d'état du GESTIONNAIRE DE TRAME	170
6-1	Indicateur simple	176
6-2	Gestionnaire de signal virtuel	178
6-3	Parleur sur demande	182
6-4	Parleur transactionnel	188
6-5	Ecouteur sur demande	192
6-6	Ecouteur transactionnel	198
6-7	Indicateur multiadresse	200
6-8	Parleur multiadresse sur demande	204
6-9	Ecouteur multiadresse transactionnel	206
6-10	Gestionnaire de priorité variable	210
6-11	Gestionnaire d'écriture	214
6-12	Gestionnaire de lecture	218
6-13	Emetteur-récepteur de bus virtuel	222
6-14	Sémaphore	228
6-15	Sémaphore à vérification de signature	240
6-16	Groupe à passage de jeton	246
6-17	Gestionnaire d'écriture avec accusé de réception	252
6-18	Ecouteur transactionnel avec accusé de réception	256
6-19	Ecouteur transactionnel à verrouillage	260
6-20	Parleur transactionnel à verrouillage	266
7-1	Chronologie et forme du signal du support physique de fond de panier	274
8-1	Réseau d'adaptation de EXTCLK	282
8-2	Réseaux d'adaptation de EXTDAT	284
8-3	Connexion à la ligne BALANCE	284
8-4	Câble plat-torsadé pour le support physique d'extension à l'intérieur d'un châssis, montrant toutes les combinaisons possibles de connecteurs	298
8-5a	Un connecteur gigogne autorise le retrait de la carte de circuit imprimé tout en assurant la continuité galvanique	300
8-5b	Câble double dans un connecteur D-sub unique assurant la continuité galvanique	302
8-6	Utilisation d'une carte de circuit imprimé pour relier le support physique d'extension en dehors et à l'intérieur d'un châssis	304
8-7	Chronologie et forme de signal pour le support physique d'extension	308
8-8	Chronologie du support physique d'extension et du fond de panier	312
8-9	Etats d'un signal d'extension de bus	316

Figure	Page
4-6 Backplane Medium waveforms	91
4-7 Test load circuit for RULE 4.26 and RULE 4.27	101
4-8 Test load circuit for RULE 4.28 and RULE 4.29	103
5-1 HEADER SENDER with FRAME MONITOR	113
5-2 HEADER SENDER state diagram	121
5-3 Signals used by a HEADER RECEIVER	127
5-4 HEADER RECEIVER state diagram	131
5-5 DATA SENDER with HEADER RECEIVER	139
5-6 DATA SENDER state diagram	147
5-7 DATA RECEIVER with HEADER RECEIVER	153
5-8 DATA RECEIVER state diagram	159
5-9 FRAME MONITOR state diagram	171
6-1 Simple Flag	177
6-2 Virtual Signal Controller	179
6-3 On-Demand Talker	183
6-4 Transaction Talker	189
6-5 On-Demand Listener	193
6-6 Transaction Listener	199
6-7 Multiaddress Flag	201
6-8 Multiaddress On-Demand Talker	205
6-9 Multiaddress Transaction Listener	207
6-10 Variable Priority Controller	211
6-11 Writing Controller	215
6-12 Reading Controller	219
6-13 Virtual Bus Transceiver	223
6-14 Semaphore	229
6-15 Signature-Checking Semaphore	241
6-16 Token Passing Group	247
6-17 Handshaking Writing Controller	253
6-18 Handshaking Transaction Listener	257
6-19 Locking Transaction Listener	261
6-20 Locking Transaction Talker	267
7-1 Backplane Medium waveforms and timing	275
8-1 EXTCLK termination network	283
8-2 EXTDAT termination networks	285
8-3 Connection to the BALANCE line	285
8-4 Twist-and-flat cable for Extended Medium within a cabinet, showing all the possible connector combinations	299
8-5a Piggyback D-sub connector allows PCB removal while main- taining galvanic continuity of the Extended Medium	301
8-5b Double cable in single D-sub connector maintaining galvanic continuity	303
8-6 Using a PCB to link the Extended Medium outside and inside a cabinet	305
8-7 Waveforms and timing for the Extended Medium	309
8-8 Extended and Backplane Media timing	313
8-9 States of an Extended bus signal	317

Tableaux

Pages

2-1	Signaux de l'interface couche physique/couche de liaison de données	40
2-2	Relation entre le bit commandé sur le support physique en fonction de XONE/XSTART/XSAM	42
2-3	Relation entre le bit reçu du support physique en fonction de l'état de RONE/RSTART	42
3-1	Utilisation des sous-trames par les modules	54
3-2	Codes de type de trame	60
3-3	Codes de la trame d'état	70
4-1	Protocole de sortie de SERDAT* pour le module d'ACCES AU BUS	94
4-2	Protocole d'entrée de SERDAT* pour le module d'ACCES AU BUS	96
7-1	REGLES de chronologie pour le groupe GENERATEUR D'HORLOGE+EXTENSION du châssis unique	274
7-2	REGLES de chronologie et OBSERVATIONS pour les modules d'ACCES AU BUS	276
7-3	Points de mesure des paramètres	280
8-1	Taux de transmission du support physique d'extension en fonction du temps de propagation et de la longueur	306
8-2	REGLES de chronologie pour un GENERATEUR D'HORLOGE dans les configurations étendues	310
8-3	REGLES et OBSERVATIONS de chronologie, groupe EXTENSION +ACCES AU BUS isolé	310
8-4	REGLES et OBSERVATIONS de chronologie pour le fond de panier/EXTENSION étendu	314
8-5	Points de mesure des paramètres	318

Table

Page

2-1	Physical Layer/Link Layer interface signals	41
2-2	XONE/XSTART/XJAM versus bit driven on medium	43
2-3	Bit received from medium versus RONE/RSTART state	43
3-1	Subframe usage by modules	55
3-2	Frame Type codes	61
3-3	Frame Status codes	71
4-1	SERDAT* output protocol for BUS ACCESS module	95
4-2	SERDAT* input protocol for BUS ACCESS module	97
7-1	Timing RULES for single-subrack CLOCK SOURCE+BRIDGE group	275
7-2	Timing RULES and OBSERVATIONS for BUS ACCESS modules	277
7-3	Parameter measurement points	281
8-1	Extended Medium data rate versus propagation versus length	307
8-2	Timing RULES for CLOCK SOURCE in extended configurations	311
8-3	Timing RULES and OBSERVATIONS, Stand-alone BRIDGE+BUS ACCESS group	311
8-4	Timing RULES and OBSERVATIONS for Backplane/Extended BRIDGE	315
8-5	Parameter measurement points	319

IECNORM.COM : Click to view the full PDF of IEC 823:1990

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

BUS SYSTEME A MICROPROCESSEURS (VMSbus) - BUS SOUS-SYSTEME SERIE DU BUS CEI 821 (VMEbus)

PREAMBULE

- 1) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par des Comités d'Etudes où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
- 2) Ces décisions constituent des recommandations internationales et sont agréées comme telles par les Comités nationaux.
- 3) Dans le but d'encourager l'unification internationale, la CEI exprime le vœu que tous les Comités nationaux adoptent dans leurs règles nationales le texte de la recommandation de la CEI, dans la mesure où les conditions nationales le permettent. Toute divergence entre la recommandation de la CEI et la règle nationale correspondante doit, dans la mesure du possible, être indiquée en termes clairs dans cette dernière.
- 4) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand il est déclaré qu'un matériel est conforme à l'une de ses recommandations.

PREFACE

La présente norme a été établie par le JTC 1/SC 26*: Système à microprocesseurs.

Le texte de cette norme est issu des documents suivants:

Règle des Six Mois	Rapport de vote	Procédure des Deux Mois	Rapport de vote
47B(BC)28	JTC 1/SC 47B N 14	JTC 1/SC 47B N 15	JTC 1/SC 47B N 20

Les rapports de vote indiqués dans le tableau ci-dessus donnent toute information sur le vote ayant abouti à l'approbation de cette norme.

La publication suivante de la CEI est citée dans la présente norme:

Publication n° 821 (1987): BUS CEI 821 - Bus système à microprocesseurs pour données de 1 à 4 octets.

* Anciennement Sous-Comité 47B du Comité d'Etudes n° 47 de la CEI.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

MICROPROCESSOR SYSTEM BUS (VMSbus) -
SERIAL SUB-SYSTEM BUS OF THE IEC 821 BUS (VMEbus)

FOREWORD

- 1) The formal decisions or agreements of the IEC on technical matters, prepared by Technical Committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules in so far as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.
- 4) The IEC has not laid down any procedure concerning marking as an indication of approval and has no responsibility when an item of equipment is declared to comply with one of its recommendations.

PREFACE

This standard has been prepared by JTC 1/SC 26*: Microprocessor systems.

The text of this standard is based on the following documents:

Six Months' Rule	Report on Voting	Two Months' Procedure	Report on Voting
47B(C0)28	JTC 1/SC 47B N 14	JTC 1/SC 47B N 15	JTC 1/SC 47B N 20

Full information on the voting for the approval of this standard can be found in the Voting Reports indicated in the above table.

The following IEC publication is quoted in this standard:

Publication No. 821 (1987): IEC 821 BUS - Microprocessor system bus for 1 to 4 byte data.

* Formerly Sub-Committee 47B of IEC Technical Committee No. 47.

BUS SYSTEME A MICROPROCESSEURS (VMSbus) -
BUS SOUS-SYSTEME SERIE DU BUS CEI 821 (VMEbus)

CHAPITRE 0: DOMAINE D'APPLICATION

La présente norme décrit une méthode d'interfaçage utilisée pour connecter des cartes à l'intérieur d'un châssis, ainsi qu'une méthode d'interfaçage étendue pour connecter entre eux des châssis et des cartes individuelles sur des distances limitées. L'interconnexion de ces cartes et châssis est effectuée par l'intermédiaire d'un support physique de données série.

IECNORM.COM : Click to view the full PDF of IEC 823:1998

MICROPROCESSOR SYSTEM BUS (VMSbus) -
SERIAL SUB-SYSTEM BUS OF THE IEC 821 BUS (VMEbus)

CHAPTER 0: SCOPE

This standard describes an interfacing method used to connect boards within a subrack, and an extended interfacing method used to connect subracks and stand-alone boards over limited distances. The interconnection of these boards and subracks is provided by a serial data medium.

IECNORM.COM : Click to view the full PDF of IEC 823:1990

CHAPITRE 1: INTRODUCTION A LA NORME CEI 823 VMSbus

1.1 *Objectifs du bus série*

Le bus série a été conçu avec les objectifs suivants:

- a) Supporter les systèmes distribués de traitement d'information dans les deux configurations correspondant à un couplage serré (multi-processeur) ou à un couplage faible.
- b) Pourvoir à la transmission rapide de messages courts et urgents entre les cartes et les châssis de tels systèmes.
- c) Supporter les systèmes tolérant aux fautes en fournissant une voie de commande qui soit complémentaire de celle du bus parallèle du fond de panier.
- d) Fournir des possibilités générales de commande et de transfert de messages incluant les opérations de diffusion, d'adressage par groupes et de scrutation de groupes.
- e) Fournir un flux de données maniable incluant des fonctions de lecture et de transfert ainsi que de la fonction plus commune d'écriture.
- f) Maximiser la disponibilité du support de transmission et la simplicité du matériel en incluant dans le processus de transmission de trame un mécanisme d'arbitrage et d'allocation.
- g) Diminuer les coûts du matériel pour les cartes qui ne nécessitent pas le taux de transmission de données d'un bus de fond de panier parallèle en éliminant l'interface au bus.
- h) Pour le bus étendu, minimiser les coûts du matériel pour les fonctions qui peuvent être réalisées sur une carte déportée simple, en supprimant les fonds de panier et les châssis.

1.1.1 *Objectifs de la norme CEI 823 VMSbus*

Les objectifs de la présente norme sont les suivants:

- a) Spécifier les caractéristiques requises pour concevoir les cartes et les circuits intégrés qui pourront communiquer de manière fiable et non ambiguë avec d'autres dispositifs connectés sur le bus série.
- b) Spécifier les protocoles qui définissent avec précision l'interaction entre le bus série et les dispositifs qui lui sont reliés.
- c) Fournir une terminologie et des définitions qui décrivent le protocole du système.

CHAPTER 1: INTRODUCTION TO THE IEC 823 VMSbus STANDARD

1.1 *Serial bus objectives*

The serial bus has been conceived with the following objectives:

- a) To support distributed information systems in both tightly coupled (multiprocessor) and loosely coupled configurations.
- b) To provide for the prompt delivery of short, urgent messages among boards and subracks in such systems.
- c) To support fault-tolerant systems by providing a control path that is complementary to a parallel backplane bus.
- d) To provide generalized control and message-transfer capabilities including broadcast, group addressing and group polling operations.
- e) To provide flexible data flow including read and move functions as well as the more common write function.
- f) To maximize medium availability and hardware simplicity by including an arbitration and allocation mechanism in the frame transmission process.
- g) To minimize hardware costs for boards that do not need the data rate of a parallel backplane bus, by eliminating the bus interface.
- h) For the extended bus, to further minimize hardware costs for functions that can be accomplished on a single remote board, by eliminating backplanes and subracks.

1.1.1 *IEC 823 VMSbus standard objectives*

This standard has the following objectives:

- a) To specify the characteristics required to design boards and integrated circuits that will reliably and unambiguously communicate with other devices attached to the serial bus.
- b) To specify protocols that precisely define the interaction between the serial bus and devices interfaced to it.
- c) To provide terminology and definitions that describe system protocol.

1.2 *Éléments du système constituant l'interface du bus série*

Cette section définit quelques termes fondamentaux utilisés pour décrire le bus série. Les termes suivants décrivent les éléments *mécaniques* des systèmes de bus série:

Connecteur

Dispositif physique qui peut être accouplé avec un autre élément semblable en vue de conduire un ou plusieurs signaux.

Fond de panier

Assemblage électronique rigide comportant des connecteurs et des pistes conductrices qui transfère les signaux d'un connecteur à l'autre en formant un bus.

Carte

Assemblage électronique rigide comportant des composants actifs, des pistes conductrices et un ou plusieurs connecteurs, qui peut être enfiché dans un fond de panier.

Emplacement

Position où une carte peut être enfichée dans un fond de panier.

Châssis

Structure rigide qui fournit un support mécanique pour les cartes enfichées dans un fond de panier, aidant à assurer que les connecteurs s'accouplent convenablement et que les cartes adjacentes n'entrent pas en contact entre elles.

Les termes suivants décrivent les éléments *fonctionnels* des systèmes de bus série:

Module fonctionnel

Ensemble conceptuel de circuits électroniques qui résident sur une carte et concourent à l'accomplissement d'une fonction spécifique. La taille et la complexité des modules est choisie de manière à permettre une spécification claire et relativement simple de leur fonction et de leur fonctionnement.

Groupe de modules

Ensemble de modules fonctionnels qui résident sur une carte et collaborent en vue d'exécuter une fonction plus élaborée.

Couche de liaison

Partie du système d'interface au bus série qui est indépendante de la configuration physique du bus et de la manière dont les signaux sont modulés sur le bus. La couche de liaison fournit des moyens fonctionnels et de procédure pour transmettre des trames. La couche de liaison est placée entre la couche physique et les couches supérieures du système.

1.2 Serial bus Interface system elements

This section defines some basic terms used to describe the serial bus. The following terms describe *mechanical* elements of serial bus systems:

Connector

A physical device which can be mated with another similar device to conduct one or more signal(s).

Backplane

A rigid electronic assembly including connectors and conductive paths that bus the signals among them.

Board

A rigid electronic assembly including active components, conductive paths, and one or more connectors that can be plugged into a backplane.

Slot

A position where a board can be plugged into a backplane.

Subrack

A rigid framework that provides mechanical support for boards inserted into a backplane, helping to ensure that the connectors mate properly and that adjacent boards do not contact each other.

The following terms describe *functional* elements of serial bus systems:

Functional module

A conceptual collection of electronic circuitry that resides on one board and works together to accomplish a specific function. The size and complexity of modules is chosen to permit clear and relatively simple specification of their function and operation.

Module group

A collection of functional modules that resides on one board and works together to accomplish a more extensive function.

Link Layer

The part of the serial bus interface system which is independent of the physical configuration of the bus and of the way that signals are modulated on the bus. The Link Layer provides functional and procedural means to transfer frames. The Link Layer stands between the Physical Layer and higher layers of the system.

Couche physique

Partie du système d'interface du bus série qui varie selon la manière dont les signaux sont modulés sur le bus. La couche physique fournit des moyens fonctionnels et de procédure pour transférer des bits. Elle comprend les spécifications électriques pour les émetteurs et récepteurs de bus. Cette norme décrit deux couches physiques. La couche physique du fond de panier utilise des niveaux de signal compatible TTL pour transférer les bits entre les cartes à l'intérieur d'un châssis. La couche physique étendue utilise une double transmission par mode différentiel pour transférer des bits entre de tels châssis et/ou des cartes individuelles.

Support physique

Parties du système d'interface du bus série qui varient selon la configuration physique du bus. Le support physique interconnecte les couches physiques des différentes cartes. Cette norme définit un support physique sur le fond de panier du bus CEI 821 VMEbus, qui relie les couches physiques des cartes à l'intérieur d'un châssis, et un support physique étendu qui relie les couches physiques entre châssis et cartes autonomes. Le support physique impose les spécifications mécaniques et les spécifications de performances.

Les termes suivants décrivent les éléments du *protocole* du bus série:

Sous-trame

Séquence de bits consécutifs sur le bus série organisée selon une structure ou un format spécifié.

Trame

Suite de sous-trames consécutives sur le bus série organisée selon une structure ou un format spécifié.

Sémaphore

En général, mécanisme qui commande l'accès à une ressource qui peut être partagée entre plusieurs utilisateurs (par exemple plusieurs processeurs). Ce terme est utilisé dans cette norme pour désigner un groupe de modules réalisant un mécanisme à bit unique qui restreint l'accès d'une ressource à un seul utilisateur à la fois et qui n'indique pas le nombre d'utilisateurs en attente de pouvoir accéder à la ressource.

Jeton

Dans une autre méthode de gestion d'une ressource partagée, un jeton constitue un droit à l'utilisation de la ressource, qui se passe d'un utilisateur à un autre. Le bus série peut être utilisé pour réaliser une gestion par jetons, mais il n'est pas lui-même géré par cette méthode.

Physical layer

The part of the serial bus interface system which varies with the way signals are modulated on the bus. The Physical Layer provides functional and procedural means to transfer bits. It includes electrical specifications for the bus drivers and receivers. This standard describes two Physical Layers. The Backplane Physical Layer uses TTL-compatible signal levels to transfer bits among boards within a subrack. The Extended Physical Layer uses dual-differential signalling to transfer bits among such subracks and/or free-standing boards.

Medium

The parts of the serial bus interface system which vary with the physical configuration of the bus. The Medium interconnects the Physical Layers on various boards. This standard defines an IEC 821 VMEbus Backplane Medium which interconnects Physical Layers on boards within a subrack, and an Extended Medium which interconnects Physical Layers between subracks and free-standing boards. The Medium dictates mechanical and performance specifications.

The following terms describe elements of the serial bus *protocol*:

Subframe

A series of consecutive bits on the serial bus in a specified arrangement or format.

Frame

A series of consecutive subframes on the serial bus in a specified arrangement or format.

Semaphore

In general, a mechanism that controls access to a resource which can be shared among several users (e.g. several processors). Used in this specification to indicate a module group that implements a single-bit mechanism which restricts access to the resource to a single user and does not indicate how many users are waiting to use the resource.

Token

In another control method for a shared resource, a token is a right to use the resource, which is passed around among users of the resource. The serial bus can be used to implement tokens, but is not itself controlled by this method.

1.3 Terminologie relative à la norme du bus série

Une norme comme celle-ci comprend une grande variété de types d'informations. D'un côté, les exigences de conformité doivent être rendues explicites, alors que d'un autre côté il est nécessaire d'inclure divers types d'information non directement liés à la conformité. Pour différencier ces types d'informations, plusieurs paragraphes de cette norme sont désignés par les MOTS CLES suivants:

REGLE
RECOMMANDATION
SUGGESTION
OBSERVATION
AUTORISATION

Tout texte non désigné avec l'un de ces MOTS CLES décrit la structure ou le fonctionnement du bus série. Il est facile à reconnaître grâce à son style narratif et l'utilisation exclusive du temps présent.

L'utilisation de ces MOTS CLES est la suivante:

REGLE chapitre.numéro:

Les REGLES constituent la structure de base de cette norme. Elles sont quelquefois exprimées sous forme textuelle, et parfois sous forme de figures, tableaux ou diagrammes d'état. Toutes les règles relatives au bus série DOIVENT être respectées pour assurer la compatibilité entre diverses réalisations sur le bus série. Les règles se caractérisent par un style impératif. Les mots DOIVENT et NE DOIVENT PAS, en lettres majuscules, sont réservés exclusivement à l'énoncé de règles dans cette norme et ne sont utilisés pour aucun autre objectif.

RECOMMANDATION chapitre.numéro:

Lorsqu'une RECOMMANDATION apparaît, les concepteurs seraient bien inspirés de suivre les conseils donnés. Opérer différemment pourrait conduire à certains problèmes graves ou à de mauvaises performances.

SUGGESTION chapitre.numéro:

Dans cette norme, une SUGGESTION contient un conseil utile mais non vital. On encourage le lecteur à considérer ce conseil avant de le rejeter. L'utilité d'une SUGGESTION dépend souvent de l'application du bus série envisagée par le concepteur.

OBSERVATION chapitre.numéro:

Les OBSERVATIONS n'offrent aucun conseil spécifique. Elles découlent en général naturellement de ce qui a été exposé auparavant. Elles explicitent les implications de certaines règles et attirent l'attention sur les points qui autrement pourraient être négligés. Elles donnent aussi le raisonnement sous-jacent à certaines REGLES, de manière que le lecteur comprenne pourquoi la REGLE doit être respectée.

AUTORISATION chapitre.numéro:

Dans certains cas, une REGLE du bus série n'interdit pas de manière spécifique une certaine approche de conception, mais le lecteur peut se demander si celle-ci peut éventuellement violer l'esprit de la REGLE, ou si elle peut conduire à quelque problème subtil. Les AUTORISATIONS rassurent le lecteur sur le fait qu'une certaine approche est acceptable, et ne causera pas de problème. Le mot PEUT, en lettres majuscules, est utilisé exclusivement pour formuler les PERMISSIONS.

1.3 *Serial bus standard terminology*

A standard like this one includes a variety of types of information. On one hand, the requirements for compliance must be spelled out, while on the other hand various types of useful data must be included which are not directly related to compliance. To differentiate these types of information, many of the paragraphs in this standard are labelled with the following KEYWORDS:

RULE
RECOMMENDATION
SUGGESTION
OBSERVATION
PERMISSION

Any text not labelled with one of these KEYWORDS describes the serial bus structure or operation. It is easily recognizable by its narrative style and exclusive use of the present tense.

These KEYWORDS are used as follows:

RULE chapter.number:

RULES form the basic framework of this standard. They are sometimes expressed in text form and sometimes in the form of figures, tables, or state diagrams. All serial bus rules MUST be followed to ensure compatibility among serial bus designs. Rules are characterized by an imperative style. The upper-case words MUST and MUST NOT are reserved exclusively for stating rules in this standard and are not used for any other purpose.

RECOMMENDATION chapter.number:

Whenever a RECOMMENDATION appears, designers would be wise to take the advice offered. Doing otherwise may result in certain awkward problems or poor performance.

SUGGESTION chapter.number:

In this standard, a SUGGESTION contains advice which is helpful but not vital. The reader is encouraged to consider the advice before discarding it. Whether or not a SUGGESTION is useful often depends on the application of the serial bus that the designer is pursuing.

OBSERVATION chapter.number:

OBSERVATIONS do not offer any specific advice. They usually follow naturally from what has been discussed previously. They spell out the implications of certain rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain RULES, so that the reader understands why the RULE must be followed.

PERMISSION chapter.number:

In some cases, a serial bus RULE does not specifically prohibit a certain design approach, but the reader might be left wondering whether that approach might violate the spirit of the RULE, or whether it might lead to some subtle problem. PERMISSIONS reassure the reader that a certain approach is acceptable, and will cause no problems. The upper-case word MAY is used exclusively for stating PERMISSIONS.

1.4 Organisation de la norme

Les chapitres suivants sont arrangés de manière à procéder à partir des principes généraux vers les détails spécifiques. Le chapitre 2 donne une vue générale du sujet. Le chapitre 3 décrit les types de trames qui apparaissent sur le bus série, et contient des détails spécifiques sur les sous-trames utilisées pour construire les trames.

Le chapitre 4 définit la couche physique, incluant les modules fonctionnels du support physique de fond de panier du bus CEI 821 VMEbus et le support physique étendu. Le chapitre 5 définit de manière formelle la couche de liaison en spécifiant le fonctionnement des modules de cette couche. Le chapitre 6 définit le protocole du bus série concernant divers groupes de modules.

Les chapitres 7 et 8 contiennent les spécifications mécaniques et de performances pour le support physique du fond de panier du bus CEI 821 VMEbus et le support physique étendu.

1.5 Relations entre le bus série et un bus parallèle du fond de panier

Lorsque le bus série est utilisé avec un bus système, il n'existe aucune contrainte de chronologie spécifiée entre un quelconque de ses signaux et n'importe lequel des signaux sur le fond de panier.

La conformité avec cette norme est indépendante de la conformité avec toute autre spécification telle que celle relative au bus CEI 821 VMEbus. Une carte ou un produit qui est construit de manière à être conforme à cette norme ne peut prétendre être compatible avec le bus CEI 821 VMEbus à moins qu'il ne contienne également une interface au bus CEI 821 VMEbus.

1.4 *Standard organization*

The following chapters are arranged to proceed from general principles to specific details. Chapter 2 gives an overview of the subject. Chapter 3 describes the types of frames that appear on the serial bus, and contains specific details of the subframes from which frames are built.

Chapter 4 defines the Physical Layer, including the functional modules for both the IEC 821 VMEbus Backplane Medium and the Extended Medium. Chapter 5 formally defines the Link Layer by specifying the operation of Link Layer modules. Chapter 6 defines the serial bus protocol of various module groups.

Chapters 7 and 8 contain the mechanical and performance specifications for the IEC 821 VMEbus Backplane Medium and the Extended Medium, respectively.

1.5 *Relationship of the serial bus and a parallel backplane bus*

When the serial bus is used with a system bus, there are no prescribed timing constraints between any of its signals and any of the other signals on the backplane.

Compliance with this standard is independent of compliance with any other standard such as that for the IEC 821 VMEbus. A board or product which is built to comply with this standard should not claim compatibility with the IEC 821 VMEbus unless it also contains an IEC 821 VMEbus interface.

CHAPITRE 2: GENERALITES SUR LE BUS SERIE

Cette norme couvre quatre parties du système d'interface du bus série, la couche de liaison de données, la couche physique, le support physique de fond de panier et le support physique étendu. La figure 2-1, page 30, montre la configuration générale d'un bus série étendu, les couches étant empilées verticalement et les cartes et châssis interconnectés au bus série répartis horizontalement.

Dans la couche de liaison de données, les modules fonctionnels sont associés en groupes formant des entités d'échange d'information sur le bus série. Les modules de la couche physique sont plus indépendants et ne sont groupés que pour deux cas spéciaux. Puisque le support physique représente simplement les liaisons électriques, il ne comporte pas de modules fonctionnels.

OBSERVATION 2.1:

La présentation en couches utilisée dans cette norme est semblable à celle qui est mise en avant dans le modèle d'interconnexion des systèmes ouverts (OSI) de l'Organisation Internationale de Normalisation (ISO). Les couches du bus série correspondent aux deux couches fonctionnelles les plus basses de OSI, le support physique en plus.

Sur la figure 2-1, les différentes interfaces du système de bus série sont représentées en lignes pointillées. La figure 2-2, page 32, fournit des détails supplémentaires sur ces interfaces pour une carte typique du bus série connectée au support physique de fond de panier.

Les interfaces entre la couche de liaison de données et les couches de niveau plus élevé de gestion sont décrites dans le chapitre 5, dans le cadre de la définition formelle des divers modules de la couche de liaison de données. Le chapitre 5 définit aussi l'interface entre les modules de la couche de liaison de données d'un même groupe. L'interface entre la couche de liaison de données et la couche physique et l'interface entre la couche physique et le support physique sont décrites d'une manière générale ci-dessous et plus en détail dans le chapitre 4.

CHAPTER 2: SERIAL BUS OVERVIEW

This standard covers four parts of the serial bus interface system, the Link Layer, the Physical Layer, the Backplane Medium, and the Extended Medium. Figure 2-1, page 31, shows the general configuration of an Extended serial bus, with the layers arranged vertically, and the boards and subracks that are interconnected by the serial bus arranged horizontally.

In the Link Layer, functional modules are arranged in groups, which are the entities that exchange information on the serial bus. Physical Layer modules are more independent and are grouped only in two special circumstances. Since the Medium simply represents conductive paths, there are no functional modules in the Medium.

OBSERVATION 2.1:

The representation of this standard in a layered form is similar to that put forth by the Open System Interconnection (OSI) model of the International Organization for Standardization (ISO). The serial bus layers correspond to the lowest two functional layers of OSI plus its physical medium.

Figure 2-1 depicts the various *interfaces* in the serial bus system as broken lines. Figure 2-2, page 33, shows further details of these interfaces, for a typical serial bus board connected to the Backplane Medium.

The interfaces between the Link Layer and higher layers of control are described in Chapter 5, as part of the formal definition of the various Link Layer modules. Chapter 5 also defines the interface between the Link Layer modules in a group. The interface between the Link Layer and the Physical Layer, and the interface between the Physical Layer and the Medium, are described in general terms below and in greater detail in Chapter 4.

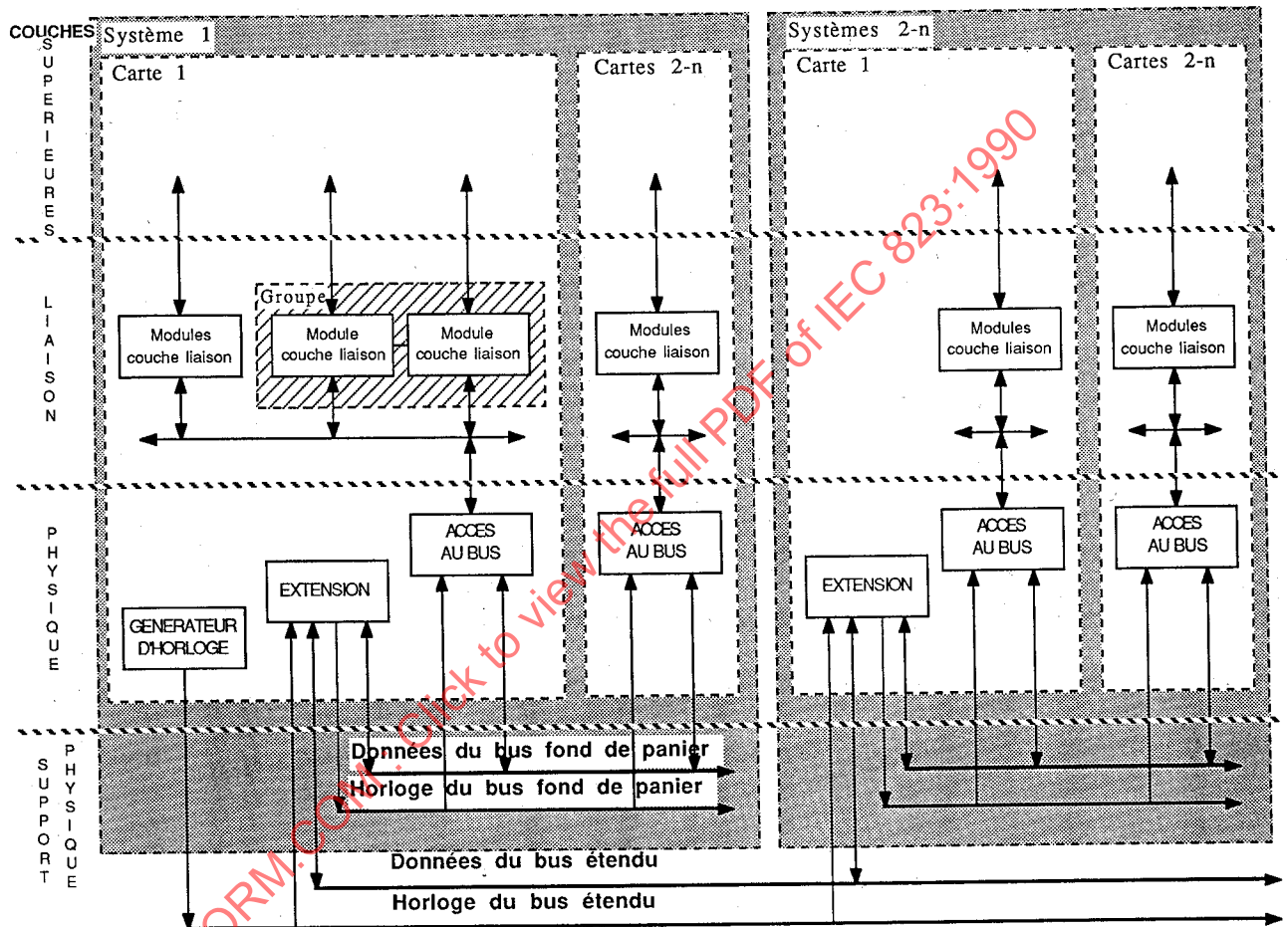


Figure 2-1 - Présentation en couches et découpage du système de bus série

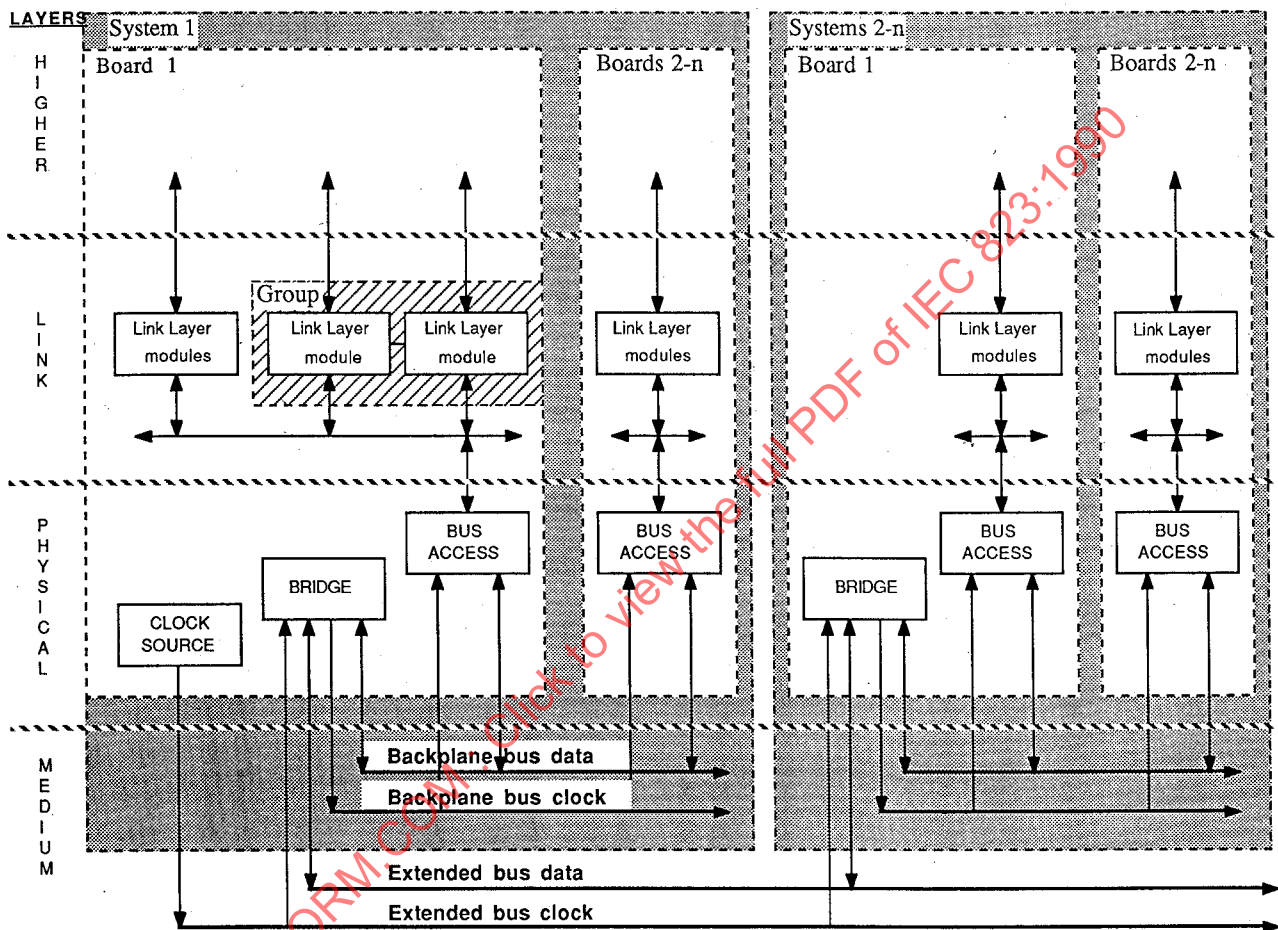


Figure 2-1 - Serial bus system structure and layering

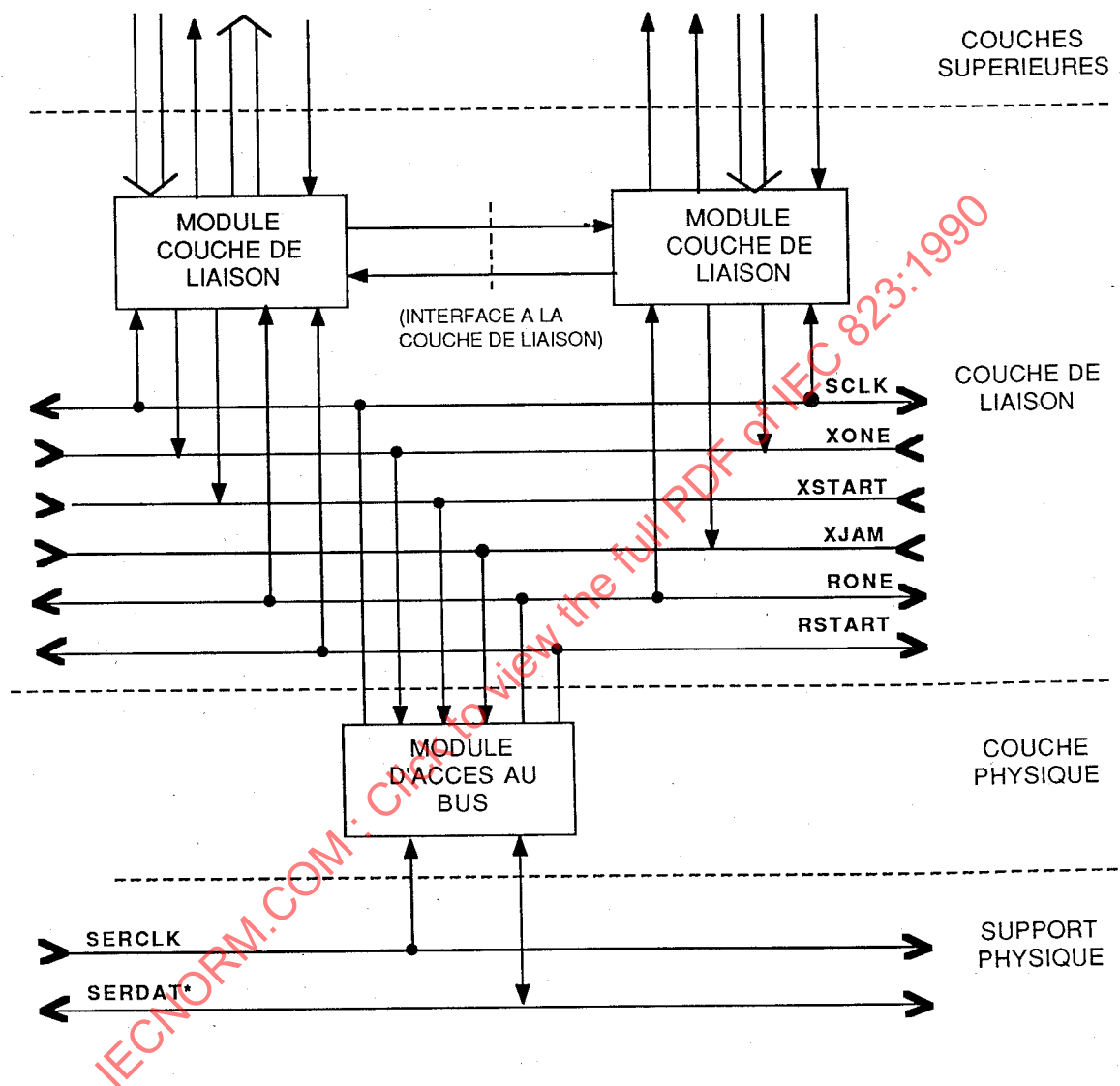


Figure 2-2 - Présentation en couches et interfaçage sur une carte typique

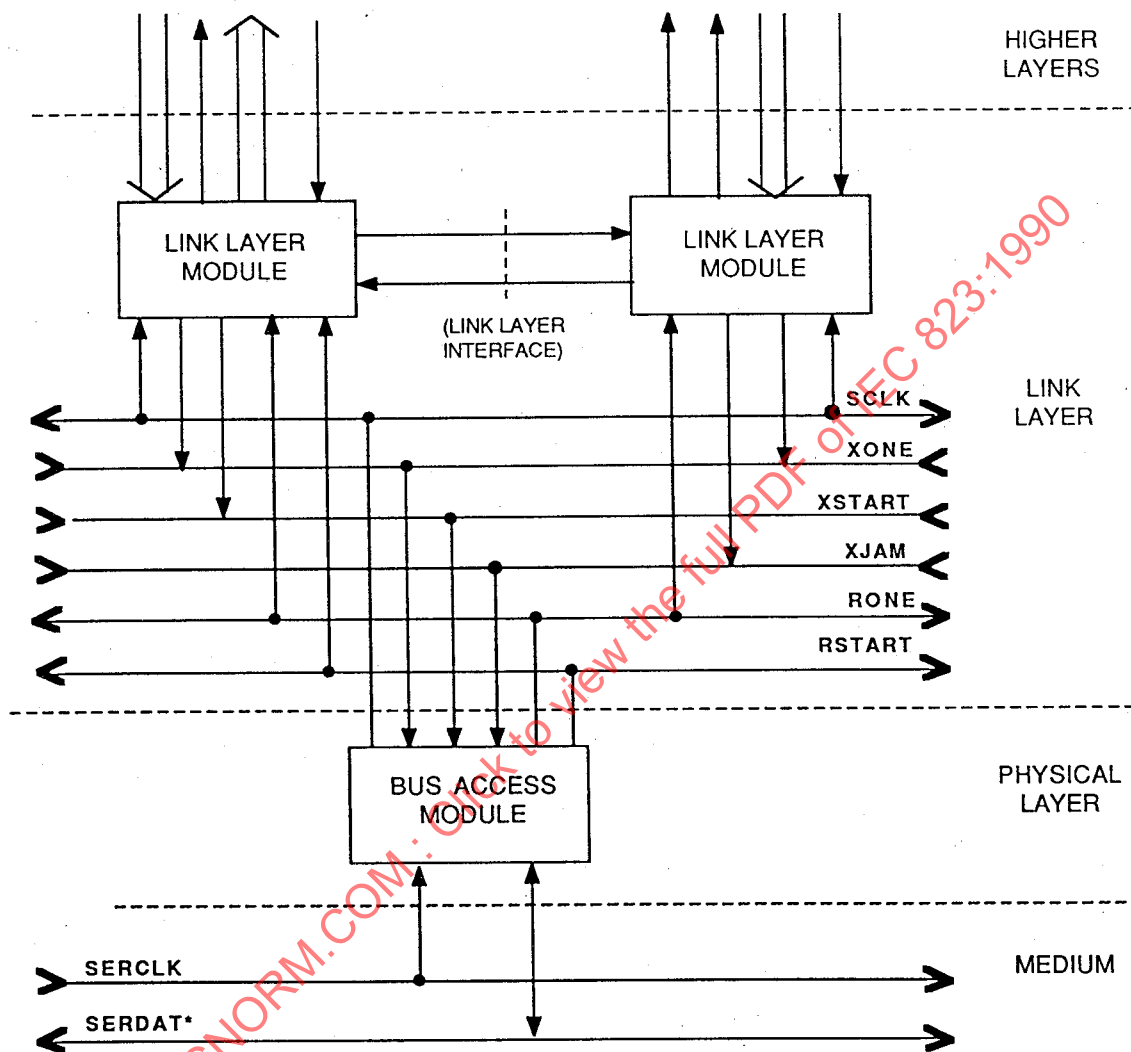


Figure 2-2 - Layering and Interfacing on a typical board

2.1 Généralités sur la couche physique

2.1.1 Signaux et modules de la couche physique

La couche physique comprend trois types de modules appelés GÉNÉRATEUR D'HORLOGE, EXTENSION et ACCES AU BUS.

Le GÉNÉRATEUR D'HORLOGE commande la paire de signaux du support physique étendu appelés EXTCLK+ et EXTCLK-. Il n'y a qu'un seul GÉNÉRATEUR D'HORLOGE (actif) sur un support physique étendu.

Le module d'EXTENSION se situe entre le support physique étendu et le support physique de fond de panier. Il y a un module d'EXTENSION (actif) pour chaque châssis ou chaque carte autonome connectés au support physique étendu. Il reçoit EXTCLK+ et EXTCLK- du support physique étendu; à partir de ces signaux il extrait et commande le signal SERCLK sur le support physique de fond de panier. Le module d'EXTENSION transfère aussi les données dans les deux sens entre la paire de signaux du support physique étendu EXTDAT+ et EXTDAT- et le signal du support physique de fond de panier SERDAT*.

Le module d'ACCES AU BUS se situe entre le support physique de fond de panier et la couche de liaison de données. Il y a un module d'ACCES AU BUS sur chaque carte échangeant des informations sur le bus série. Il reçoit SERCLK et commande et reçoit SERDAT*.

OBSERVATION 2.2:

Les lignes EXTCLK+ et EXTCLK- transportent un signal logique sous forme complémentaire (inversee). Quand la distinction entre les deux lignes n'est pas nécessaire, cette norme se réfère au signal logique sous l'appellation EXTCLK. De même EXTDAT+ et EXTDAT- transportent un signal logique appelé EXTDAT.

Dans la configuration complète de bus série donnée en exemple dans la figure 2-1, page 30, EXTCLK et EXTDAT relient les châssis et les cartes autonomes tandis que SERCLK et SERDAT* relient les cartes à l'intérieur d'un châssis. Toutefois deux autres cas de configuration du bus série méritent aussi attention. La figure 2-3, page 36, montre une configuration à châssis unique pour laquelle les fonctions GÉNÉRATEUR D'HORLOGE et EXTENSION sont combinées en un groupe de modules fonctionnels sur la carte à l'emplacement 1, et où le support physique étendu n'est pas utilisé. La figure 2-4, page 36, montre une carte autonome sur laquelle des fonctions des modules EXTENSION et ACCES AU BUS sont combinées en un groupe, et où le support physique de fond de panier n'est pas utilisé. Ces configurations sont sujettes à des considérations particulières sur les couches physiques décrites dans les chapitres 4, 7 et 8.

2.1.2 Utilisation des signaux de la couche physique

Les signaux d'horloge (EXTCLK et SERCLK) servent à synchroniser l'activité du bus série sur les signaux de données (EXTDAT et SERDAT*). La chronologie de EXTCLK et SERCLK dépend du fait que le support physique étendu est utilisé ou non et, s'il l'est, de sa longueur.

EXTDAT et SERDAT* véhiculent les bits d'information parmi les modules d'ACCES AU BUS sur les différentes cartes du bus série. Ces bits sont de trois types: bit_un, bit_zéro et bit_départ.

2.1 Overview of the Physical Layer

2.1.1 Physical Layer modules and signals

The Physical Layer includes three types of modules called CLOCK SOURCE, BRIDGE and BUS ACCESS.

The CLOCK SOURCE drives the Extended Medium signal pair called EXTCLK⁺ and EXTCLK⁻. There is only one (active) CLOCK SOURCE on an Extended Medium.

The BRIDGE stands between the Extended Medium and a Backplane Medium. There is one (active) BRIDGE for each subrack or stand-alone board connected to the Extended Medium. It receives EXTCLK⁺ and EXTCLK⁻ from the Extended Medium; based on these it derives and drives the SERCLK signal on the Backplane Medium. The BRIDGE also passes data bidirectionally between the Extended Medium signal pair called EXTDAT⁺ and EXTDAT⁻, and the Backplane Medium signal SERDAT*.

The BUS ACCESS module stands between a Backplane Medium and the Link Layer. There is one BUS ACCESS module on each board that communicates on the serial bus. It receives SERCLK and drives and receives SERDAT*.

OBSERVATION 2.2:

The EXTCLK⁺ and EXTCLK⁻ lines carry one logical signal in complementary (inverted) form. When there is no need to differentiate between the two lines, this standard will refer to the logical signal as EXTCLK. Similarly, EXTDAT⁺ and EXTDAT⁻ carry one logical signal that is referred to as EXTDAT.

In the full-scale serial bus configuration exemplified by Figure 2-1, page 31, EXTCLK and EXTDAT interconnect subracks and stand-alone boards, while SERCLK and SERDAT* interconnect boards within a subrack. However, two other cases of serial bus configurations also deserve attention. Figure 2-3, page 37, shows a single-subrack configuration in which the CLOCK SOURCE and BRIDGE functions are combined into a functional module group on the board in Slot 1, and the Extended Medium is not used. Figure 2-4, page 37, shows a free-standing board on which the BRIDGE and BUS ACCESS functions are combined into a group, and the Backplane Medium is not used. These configurations are subject to special Physical Layer considerations as described in Chapters 4, 7 and 8.

2.1.2 Physical Layer signalling

The clock signals (EXTCLK and SERCLK) serve to synchronize serial bus activity on the data signals (EXTDAT and SERDAT*). The timing of EXTCLK and SERCLK is a function of whether the Extended Medium is used, and if so of its length.

EXTDAT and SERDAT* convey information bits among the BUS ACCESS modules on the various boards on the serial bus. These bits are of three types: one_bits, zero_bits and start_bits.

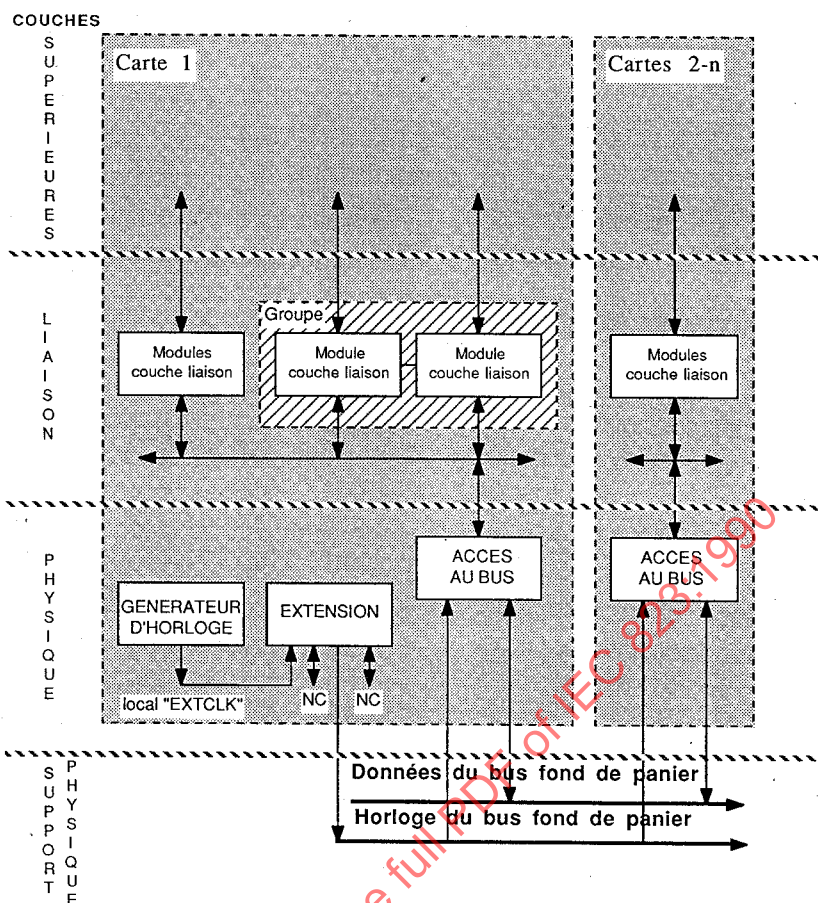


Figure 2-3 - Configuration du support physique de fond de panier pour un châssis simple

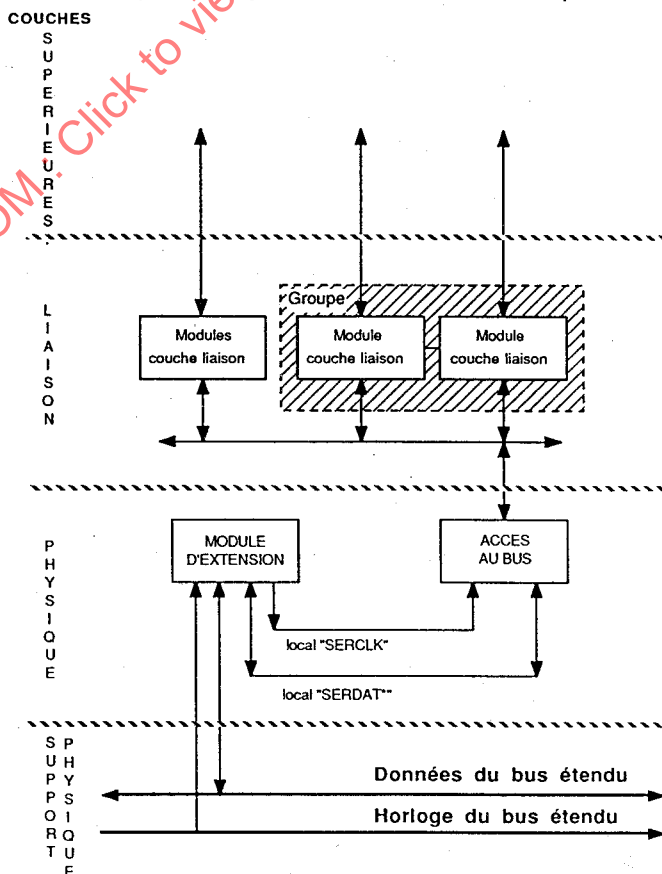


Figure 2-4 - Carte indépendante sur le support physique étendu

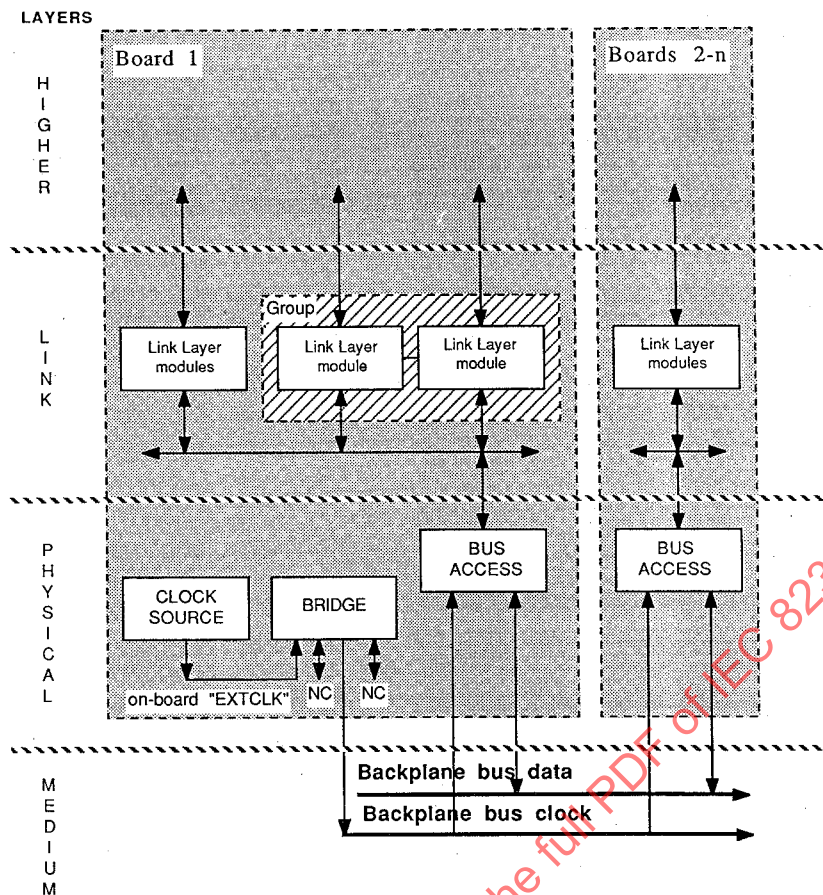


Figure 2-3 - Single-subrack backplane medium configuration

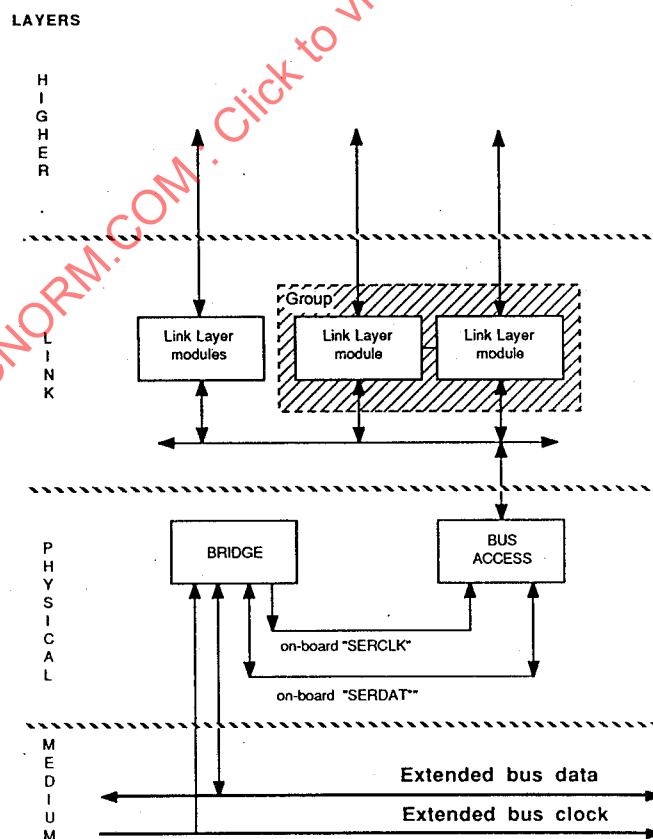


Figure 2-4 - Free-standing board on the Extended Medium

Les différents modules d'ACCES AU BUS sur un bus série peuvent présenter au même instant des bits de types différents sur SERDAT*. Lorsque le support physique étendu est utilisé, les divers modules d'EXTENSION présentent alors des bits de types différents sur EXTDAT. Dans ce cas, le bit résultant reçu par tous les modules d'ACCES AU BUS sur le bus est le même et est déterminé comme suit:

REGLE 2.1:

Les modules d'ACCES AU BUS et d'EXTENSION DOIVENT être conçus de telle sorte que, lorsque un module d'ACCES AU BUS sur le bus série présente un bit_un, tous les modules d'ACCES AU BUS sur le bus reçoivent un bit_un.

REGLE 2.2:

Les modules d'ACCES AU BUS et d'EXTENSION DOIVENT être conçus de telle sorte que si aucun module d'ACCES AU BUS sur le bus série ne présente de bit_un, mais qu'un ou plusieurs d'entre eux présentent un bit_départ, tous les modules d'ACCES AU BUS sur le bus reçoivent un bit_départ.

REGLE 2.3:

Les modules d'ACCES AU BUS et d'EXTENSION DOIVENT être conçus de telle sorte que lorsque tous les modules d'ACCES AU BUS sur le bus série présentent des bit_zéro, tous les modules d'ACCES AU BUS sur le bus reçoivent un bit_zéro.

Le chapitre 4 définit les modules de la couche physique pour le support physique du fond de panier, qui satisfont aux règles ci-dessus avec des niveaux de tension compatibles TTL. Il définit aussi les modules de la couche physique pour le support physique étendu, qui satisfait aux règles avec des signaux différentiels semblables à ceux des normes EIA (Electronic Industries Association) RS-422 et RS-485.

2.2 Généralités sur la couches de liaison de données

2.2.1 Modules et interfaces

La couche de liaison de données comprend cinq types de modules appelés EMETTEURS D'EN-TETE, GESTIONNAIRES DE TRAME, RECEPTEURS D'EN-TETE, EMETTEURS DE DONNEES et RECEPTEURS DE DONNEES. Toute combinaison de n'importe quel nombre de ces modules peut être présente sur toute carte du bus série.

OBSERVATION 2.3:

Les noms des modules ont été choisis pour rappeler leur fonction principale. Ils n'indiquent pas de restriction sur le fonctionnement de ces modules. Par exemple, les RECEPTEURS D'EN-TETE et RECEPTEURS DE DONNEES peuvent tous deux *envoyer* aussi des données sur le bus série dans le champ d'état de la Trame, comme il est décrit dans les chapitres suivants.

L'interface entre les modules de la couche de liaison de données et le module d'ACCES AU BUS de la couche physique consiste en six signaux donnés dans le tableau 2-1. Les signaux SCLK, RONE et RSTART sont commandés par le module d'ACCES AU BUS et reçus par les modules de la couche de liaison de données, tandis que les signaux XONE, XSTART et XJAM sont commandés par les modules de la couche de liaison de données et reçus par le module d'ACCES AU BUS.

The various BUS ACCESS modules on a serial bus are allowed to present different types of bits on SERDAT* at the same time. When the Extended Medium is used, the various BRIDGE modules thus present different types of bits on EXTDAT. When this occurs, the resulting bit received by all the BUS ACCESS modules on the bus is the same and is determined as follows:

RULE 2.1:

BUS ACCESS modules and BRIDGES MUST be designed so that, when any BUS ACCESS module on a serial bus presents a one_bit, all the BUS ACCESS modules on the bus receive a one_bit.

RULE 2.2:

BUS ACCESS modules and BRIDGES MUST be designed so that, when no BUS ACCESS module on a serial bus presents a one_bit, but one or more present a start_bit, all BUS ACCESS modules on the bus receive a start_bit.

RULE 2.3:

BUS ACCESS modules and BRIDGES MUST be designed so that, when all BUS ACCESS modules on a serial bus present zero_bits, all BUS ACCESS modules on the bus receive a zero_bit.

Chapter 4 defines the Physical Layer modules for the Backplane Medium, which implement the above rules using TTL-compatible voltage levels. It also defines the Physical Layer modules for the Extended Medium, which implement the rules using differential signalling similar to that used in EIA (Electronic Industries Association) standards RS-422 and RS-485.

2.2 Link Layer overview

2.2.1 Modules and Interfaces

The Link Layer includes five types of modules called HEADER SENDERS, FRAME MONITORS, HEADER RECEIVERS, DATA SENDERS and DATA RECEIVERS. Any number and combination of these modules can be present on any board on a serial bus.

OBSERVATION 2.3:

Module names are intended to indicate the major function of each module. They are not meant to constrain the modules' operation. For example, both HEADER RECEIVERS and DATA RECEIVERS can also send information on the serial bus in the Frame Status field, as described in subsequent chapters.

The interface between Link Layer modules and the Physical Layer BUS ACCESS module consists of six signals as shown in Table 2-1. The SCLK, RONE, and RSTART signals are driven by the BUS ACCESS module and received by the Link Layer modules, while XONE, XSTART, and XJAM are driven by the Link Layer modules and received by the BUS ACCESS module.

Le signal SCLK est extrait du signal SERCLK par le module d'ACCES AU BUS. Il sert à informer les modules de la couche de liaison de données de la chronologie des bits de telle sorte qu'ils puissent présenter des bits sur XONE et XSTART et interpréter les bits reçus sur RONE et RSTART. Le module d'ACCES AU BUS utilise XONE, XSTART et XJAM pour présenter les bits sur SERDAT* et présente l'information reçue de SERDAT* sur RONE et RSTART.

Les signaux XONE, XSTART, XJAM, RONE et RSTART sont commandés à l'état vrai ou faux en correspondance aux bit_un, bit_zéro, bit_départ de SERDAT*, comme indiqué dans les tableaux 2-2 et 2-3.

OBSERVATION 2.4:

L'information sur les lignes XONE, XSTART, XJAM, RONE et RSTART est équivalente à celle sur SERDAT*. Les modules de la couche de liaison de données pourraient être décrits comme utilisant SERDAT* directement, ou un signal unique identique dans l'interface couche physique/couche de liaison de données. Le modèle multilignes est utilisé parce qu'il simplifie la description formelle des modules de la couche de liaison de données.

Le bus série permet aux différents modules de la couche de liaison de données sur une carte de présenter au même moment des états différents sur XONE, XSTART et XJAM.

REGLE 2.4:

Les modules de la couche de liaison de données DOIVENT être conçus de telle sorte que, lorsqu'un module quelconque de la couche de liaison de données sur une carte présente XONE, XSTART ou XJAM à l'état vrai, le module d'ACCES AU BUS reçoive ce signal à l'état vrai même si un autre module de la couche de liaison de données sur la carte présente le signal à l'état faux.

Tableau 2-1 - Signaux de l'interface couche physique/couche de liaison de données

Signal	Commandé par la	Utilisation
SCLK	Couche physique	Donne la chronologie des bits à la couche de liaison de données
XONE	Couche de liaison de données	Ordonne à la couche physique d'envoyer un bit_un
XJAM	Couche de liaison de données	Ordonne à la couche physique d'envoyer un bit_un
XSTART	Couche de liaison de données	Ordonne à la couche physique d'envoyer un bit_départ
RONE	Couche physique	Indique à la couche de liaison de données la réception d'un bit_un
RSTART	Couche physique	Indique à la couche de liaison de données la réception d'un bit_départ

The SCLK signal is derived from the SERCLK signal by the BUS ACCESS module. It serves to inform the Link Layer modules of bit timing so that they may present bits on XONE and XSTART and interpret received bits on RONE and RSTART. The BUS ACCESS module uses XONE, XSTART, and XJAM to present bits on SERDAT*, and presents the information received from SERDAT* on RONE and RSTART.

The XONE, XSTART, XJAM, RONE, and RSTART signals are driven to True or False corresponding to one_bits, zero_bits, and start_bits on SERDAT*, as shown in Tables 2-2 and 2-3.

OBSERVATION 2.4:

The information on the XONE, XSTART, XJAM, RONE, and RSTART lines is equivalent to that on SERDAT*. Link Layer modules could be described as using SERDAT* directly, or an identical single signal in the Physical Layer/Link Layer interface. The multi-line scheme is used because it simplifies the formal description of Link Layer modules.

The serial bus allows the various Link Layer modules on a board to present different states on XONE, XSTART and XJAM at the same time.

RULE 2.4:

Link Layer modules MUST be designed so that, when any Link Layer module on a board presents XONE, XSTART, or XJAM True, the BUS ACCESS module receives that signal True, even if another Link Layer module on the board presents the signal False.

Table 2-1 - Physical Layer/Link Layer interface signals

Signal	Driven by	Use
SCLK	Physical Layer	Signals bit timing to the Link Layer
XONE	Link Layer	Commands the Physical Layer to send a one_bit
XJAM	Link Layer	Commands the Physical Layer to send a one_bit
XSTART	Link Layer	Commands the Physical Layer to send a start_bit
RONE	Physical Layer	Indicates a received one_bit to the Link Layer
RSTART	Physical Layer	Indicates a received start_bit to the Link Layer

Tableau 2-2 - Relation entre le bit commandé sur le support physique en fonction de XONE/XSTART/XJAM

XJAM	XONE	XSTART	Bit commandé sur le support physique
Vrai	x	x	bit_un
Faux	Vrai	x	bit_un
Faux	Faux	Vrai	bit_départ
Faux	Faux	Faux	bit_zéro

Note:

"x" dans le tableau ci-dessus indique que le signal est sans effet sur le bit commandé sur le support physique.

Tableau 2-3 - Relation entre le bit reçu du support physique en fonction de l'état de RONE/RSTART

Bit reçu du support physique	RONE	RSTART
bit_un	Vrai	Faux
bit_départ	Faux	Vrai
bit_zéro	Faux	Faux

2.2.2 Groupes de modules

Les modules de la couche de liaison de données sont délibérément limités en dimension et en fonctions dans cette norme de façon que les descriptions formelles de leur fonctionnement restent simples. Cependant, le fonctionnement total du bus série est plus facile à décrire en termes d'entités plus importantes appelées *groupes de modules*.

Les groupes suivants sont les plus courants:

Gestionnaire: Un EMETTEUR D'EN-TETE et un GESTIONNAIRE DE TRAME

Indicateur: Un RECEPTEUR D'EN-TETE et une bascule. Les groupes indicateurs sont classés de manière plus détaillée selon le protocole de fonctionnement, en indicateur_simple, sémaphore et groupe à passage de jeton

Parleur: Un RECEPTEUR D'EN-TETE et un EMETTEUR DE DONNEES

Ecouteur: Un RECEPTEUR D'EN-TETE et un RECEPTEUR DE DONNEES

Table 2-2 - XONE/XSTART/XJAM versus bit driven on medium

XJAM	XONE	XSTART	Bit driven on medium
True	x	x	One_bit
False	True	x	One_bit
False	False	True	Start_bit
False	False	False	Zero_bit

Note:

The "x" in the above table indicates that the signal has no effect on the bit driven on the medium.

Table 2-3 - Bit received from medium versus RONE/RSTART state

Bit received from medium	RONE	RSTART
One_bit	True	False
Start_bit	False	True
Zero_bit	False	False

2.2.2 Module groups

Link Layer modules are deliberately restricted in size and functionality in this standard, so that the formal descriptions of their operation remain straightforward. However, the overall operation of the serial bus is easier to describe in terms of larger entities called *module groups*.

The following are the most common groups:

Controller: A HEADER SENDER and a FRAME MONITOR

Flag: A HEADER RECEIVER and a latch. Flag groups are further classified according to their operational protocol, into Simple_Flag, Semaphore and Token Passing groups

Talker: A HEADER RECEIVER and a DATA SENDER

Listener: A HEADER RECEIVER and a DATA RECEIVER

2.2.3 Protocole de transmission d'une trame

Toute communication entre des groupes de modules de la couche de liaison de données s'effectue à travers la couche physique par l'envoi de trames sur le bus série. Ces trames sont composées de "sous-trames" envoyées par les divers groupes. Une trame commence quand un groupe contrôleur envoie une "sous-trame d'en-tête". Les autres groupes répondent alors en envoyant des sous-trames conformes à un protocole prescrit jusqu'à ce que la fin de la trame soit atteinte.

Le GESTIONNAIRE DE TRAME dans chaque groupe gestionnaire reçoit chacun des bits du bus série et suit l'évolution de chaque trame. Un EMETTEUR D'EN-TETE peut alors déterminer s'il y a une trame en cours, à partir d'un signal sur la carte, généré par un GESTIONNAIRE DE TRAME. Lorsqu'il n'y a aucune trame en cours, il peut en démarrer une par l'envoi d'un en-tête.

Il n'y a ni lignes de signaux séparées, ni période de temps allouée pour déterminer quels sont les modules de la couche de liaison de données qui peuvent envoyer une information sur le bus série. Un mécanisme d'arbitrage est plutôt incorporé dans la procédure de transmission des trames.

Quand un module EMETTEUR D'EN-TETE transmet un en-tête, il échantillonne chaque bit sur la ligne RONE pendant qu'il émet. Si l'EMETTEUR D'EN-TETE détecte RONE à l'état vrai alors qu'il présente XONE à l'état faux, il cesse d'émettre. Cela permet aux autres EMETTEURS D'EN-TETE de terminer l'émission de la sous-trame d'en-tête sans interférence.

OBSERVATION 2.5:

Cette méthode d'arbitrage permet à plusieurs EMETTEURS D'EN-TETE de commencer simultanément l'envoi de trames, sans affecter les transmissions des autres. (Une seule de ces transmissions se poursuivra tandis que les autres essaieront à nouveau plus tard.)

L'en-tête "gagnant" spécifie quels seront les autres modules qui participeront à la transmission de la trame, au moyen de ses dix bits "adresse S" et de ses dix bits "adresse R". Chaque RECEPTEUR D'EN-TETE sur le bus, qu'il soit dans un groupe parleur, écouteur ou indicateur, a une adresse qu'il compare aux deux champs de l'en-tête. S'il trouve une concordance sur l'un de ces champs, il répond en signalant la concordance aux autres modules dans le groupe. Selon le type de groupe auquel le RECEPTEUR D'EN-TETE appartient, la concordance d'adresse peut avoir différents effets:

- 1) dans un groupe indicateur, il peut changer l'état de la bascule;
- 2) dans un groupe parleur, il peut signaler à l'EMETTEUR DE DONNEES d'émettre, ou
- 3) dans un groupe écouteur, il peut signaler au RECEPTEUR DE DONNEES de recevoir.

2.2.3 Frame transmission protocol

All communication between Link Layer module groups is done through the Physical Layer, by sending frames on the serial bus. These frames are composed of "subframes" which are sent by various groups. A frame is initiated when a Controller group sends a "Header subframe". Other groups then respond by sending subframes according to a prescribed protocol until the end of the frame is reached.

The FRAME MONITOR in each Controller group receives every bit from the serial bus and follows the progress of each frame. A HEADER SENDER can thus determine whether there is a frame in progress, from an on-board signal generated by its FRAME MONITOR. Whenever there is no frame in progress, it can initiate one by sending a Header.

There are no separate signal lines or time period dedicated to determining which Link Layer modules may send information on the serial bus. Rather, an arbitration mechanism is built into the procedure for transmitting frames.

When a HEADER SENDER module transmits a Header, it samples each bit on the RONE line while it is sending. If the HEADER SENDER detects RONE True when it presented XONE False, it stops sending. This allows other HEADER SENDER(S) to finish sending the Header subframe without interference.

OBSERVATION 2.5:

This method of arbitration allows several HEADER SENDERS to start sending frames simultaneously, without affecting each other's transmissions. (One of the transmissions will be successful while the others will be tried again later.)

The "winning" Header specifies what other modules will participate in the frame transmission, by means of its ten-bit "S address" and ten-bit "R address". Each HEADER RECEIVER on the bus, be it in a Talker, Listener, or Flag group, has an address which it compares with these two fields in the Header. If it finds a match in either of these fields, it responds by signalling the match to the other modules in the group. Depending on which type of group the HEADER RECEIVER is in, an address match may have different effects:

- 1) in a Flag group, it may change the state of the latch;
- 2) in a Talker group, it may signal the DATA SENDER to send, or
- 3) in a Listener group, it may signal the DATA RECEIVER to receive.

2.3 Utilisation du bus série pour transférer des données

Pour transférer des données, l'EMETTEUR D'EN-TETE envoie une adresse S correspondant à un parleur sur le bus et une adresse R correspondant à un écouteur. (Les adresses réelles utilisées pour sélectionner ces groupes dépendent de la manière dont le système est configuré. Par exemple, il n'y a pas de bloc d'adresse normalisé alloué spécialement à la sélection des écouteurs.)

Chaque RECEPTEUR D'EN-TETE sur le bus série compare ces adresses avec les siennes. Dans une trame de transfert de données au moins un des RECEPTEURS D'EN-TETE des parleurs trouve une concordance avec l'adresse S. Il indique à son EMETTEUR DE DONNEES d'envoyer les données. De même, un ou plusieurs RECEPTEURS D'EN-TETE des écouteurs trouvent une concordance avec l'adresse R et indiquent à leur RECEPTEUR DE DONNEES de recevoir les données.

Le nombre réel d'octets transférés est laissé au choix de l'EMETTEUR DE DONNEES. Après que l'EMETTEUR D'EN-TETE a envoyé l'en-tête, l'EMETTEUR DE DONNEES envoie une sous-trame de trois bits indiquant le nombre d'octets qu'il a l'intention d'envoyer aux RECEPTEURS DE DONNEES, suivie par les octets de données. Ensuite, le RECEPTEUR DE DONNEES répond par un code d'état, pour indiquer qu'il a été sélectionné et s'il a été capable de recevoir les données.

2.4 Utilisation du bus série pour positionner à un et à zéro des indicateurs

Quand un EMETTEUR D'EN-TETE est utilisé pour positionner à un ou à zéro un groupe indicateur, il envoie un en-tête comme dans le cas décrit ci-dessus. Si la trame doit *positionner* un indicateur à un, il envoie l'adresse de l'indicateur dans le champ S. Si la trame doit positionner un indicateur à zéro, il envoie l'adresse de l'indicateur dans le champ R.

Quand un en-tête est envoyé, chaque RECEPTEUR D'EN-TETE sur le bus série vérifie ses adresses S et R pour voir si son groupe de modules est sélectionné. Un ou plusieurs RECEPTEURS D'EN-TETE reconnaissent leur adresse S (ou R) et positionnent à un (ou remettent à zéro) leur bascule.

De plus amples informations sur la manière dont les trames sont utilisées pour commander des groupes indicateurs sont indiquées dans le chapitre 6.

2.5 Applications des groupes de modules du bus série

Le chapitre 6 explique comment des groupes de modules de la couche de liaison de données comme ceux décrits ci-dessus peuvent être utilisés comme blocs pour construire une grande variété de configurations de systèmes. Par exemple, la combinaison d'un contrôleur, parleur et un écouteur implantés chacun sur deux cartes, peut être utilisée par une des cartes pour transmettre une adresse et lire le

2.3 *Using the serial bus to transfer data*

To transfer data, the HEADER SENDER sends an S Address that corresponds to a Talker on the bus, and an R Address that corresponds to a Listener. (The actual addresses used to select these groups depend on how the system is configured. For example, there is no standard block of addresses assigned specifically for selecting Listeners.)

Each HEADER RECEIVER on the serial bus compares these addresses against its own. In a Data Transfer frame, at least one of the Talkers' HEADER RECEIVERS finds a match with the S address. It tells its DATA SENDER to send data. Similarly, one or more of the Listeners' HEADER RECEIVERS find a match with the R address and tells its DATA RECEIVER to receive data.

The actual number of bytes transferred is left up to the DATA SENDER. After the HEADER SENDER has sent the Header, the DATA SENDER sends a three bit subframe indicating the number of bytes it intends to send to the DATA RECEIVER, followed by the data bytes. After this, the DATA RECEIVER responds with a status code, to indicate that it was selected and whether it was able to accept the data.

2.4 *Using the serial bus to set and reset flags*

When a HEADER SENDER is used to set or reset a Flag group, it sends a Header as in the case described above. If the frame is intended to set a flag, it sends the Flag's address in the S field. If the frame is intended to reset a flag, it sends the Flag's address in the R field.

When the Header is sent, each HEADER RECEIVER on the serial bus checks its S and R addresses to see if its module group is selected. One (or more) HEADER RECEIVER(S) matches the S (or R) address and sets (or resets) its latch.

Further information on how frames are used to control Flag groups is given in Chapter 6.

2.5 *Applications of serial bus module groups*

Chapter 6 explains how groups of Link Layer modules such as those described above can be used as building blocks for a wide variety of system configurations. For example, the combination of a Controller, Talker and Listener, on each of two boards, can be used by one board to pass an address and read the contents of a memory location on the

contenu d'un emplacement de mémoire sur l'autre. Les groupes indicateurs peuvent être utilisés pour réinitialiser une ou plusieurs cartes dans un système, ou pour déconnecter du bus système, de façon sélective, une carte défaillante. Les indicateurs peuvent aussi être utilisés comme des sémaphores qui peuvent avoir des avantages de fonctionnement sur les sémaphores en mémoire commune ou pour des fonctions de passage de jeton.

2.6 Possibilités d'adressage du bus série

Comme il est indiqué précédemment, les adresses S ou R d'une trame peuvent sélectionner plus d'un RECEPTEUR D'EN-TETE et donc plus d'un groupe de modules. Cela est à l'origine de quelques-unes des possibilités les plus intéressantes du bus série.

En sélectionnant plusieurs écouteurs, des fonctionnements en "diffusion" ou en "groupe d'adresse" sont possibles. Par exemple, un écouteur sur une carte processeur peut répondre à des trames adressées à tous les processeurs, à des processeurs d'un type donné ou à sa propre unique adresse.

En sélectionnant plusieurs parleurs, un fonctionnement en "scrutation de groupe" est possible. Par exemple, un parleur associé à un capteur de température pourrait envoyer des données en réponse à des entêtes qui sélectionnent "tous les capteurs de températures", "tous les capteurs de zone N" ou ce capteur spécifique.

Pour un tel fonctionnement en scrutation de groupe, les parleurs peuvent ou non utiliser l'arbitrage du bus série (similaire à celui utilisé par les EMETTEURS D'EN-TETE) pendant qu'ils envoient les données. Si l'arbitrage du champ de données est utilisé, la donnée résultante est la valeur binaire la plus grande entre les données des parleurs (par exemple la température la plus haute). Si l'arbitrage du champ de données n'est pas utilisé, la donnée résultante est un OU entre les données des parleurs, ce qui est utile pour des informations de type état (par exemple pour des bits "d'alarme").

Les fonctionnements en diffusion et en groupe d'adresses peuvent aussi s'appliquer aux groupes Indicateurs. Par exemple, si des groupes indicateurs sont utilisés pour pouvoir déconnecter des cartes défaillantes du bus système, une trame peut être envoyée pour déconnecter "tous les maîtres du bus", un maître particulier ou "toutes les cartes du fabricant XYZ".

other. Flag groups can be used to re-initialize one or more of the boards in a system, or to selectively disconnect a failed board from the system bus. Flags can also be used to provide semaphores that can have operational advantages over semaphores in a common memory, or to provide token passing functions.

2.6 Serial bus addressing capabilities

As indicated above, the S or R address in a frame can select more than one HEADER RECEIVER, and thus more than one module group. This fact provides some of the more powerful capabilities of the serial bus.

When multiple Listeners are selected, "broadcast" or "group address" operations can be done. For example, a Listener on a processor card might respond to frames addressed to all processors, processors of a specific type or its own unique address.

When multiple Talkers are selected, "group poll" operations can be done. For example, a Talker connected to a temperature sensor might send data in response to Headers which select "all temperature sensors", "all sensors in zone N" or this specific sensor.

For such group poll operations, Talkers can be commanded whether to use serial bus arbitration (like that used by HEADER SENDERS) while sending the data. If data field arbitration is used, the resultant data is the largest binary value among the Talkers' data (e.g. the highest temperature). If data field arbitration is not used, the resultant data is the OR of the Talkers' data, which is useful for status-type information (e.g. for "alarm" bits).

Broadcast and group address operations can also be applied to Flag groups. For example, if Flag groups are used to allow disconnection of failed boards from the system bus, a frame could be sent to disconnect "all bus masters", a particular bus master or "all boards made by manufacturer XYZ".

CHAPITRE 3: TRAMES ET SOUS-TRAMES DU BUS SERIE

3.1 *Types généraux de trames*

Comme l'indique le chapitre 2, une trame est composée de sous-trames envoyées en séquence par plusieurs groupes de modules, chaque fois qu'un groupe gestionnaire envoie un en-tête. En fonction des types de groupes de modules sélectionnés par l'en-tête, trois types généraux de trames peuvent apparaître sur le bus série, comme indiqué dans la figure 3-1, page 52.

Une trame annulée indique qu'un groupe de modules sélectionné n'est pas prêt à participer à la trame.

Si tous les groupes sélectionnés sont prêts et l'adresse S de l'en-tête sélectionne un groupe parleur, une trame de transfert de données en résulte. Les trames de transfert de données envoient 1-32 octets de données à partir des groupes parleurs vers des groupes écouteurs et peuvent aussi affecter des groupes indicateurs utilisés pour suivre des transferts de données.

Une trame de commande résulte si tous les groupes sélectionnés sont prêts et l'adresse S n'a pas sélectionné un groupe parleur. Les trames de commande ne contiennent aucune donnée et sont utilisées seulement pour positionner à un ou à zéro des groupes indicateurs.

Le tableau 3-1 définit quels modules commandent et reçoivent les différentes sous-trames dans chacun de ces types généraux de trame.

Puisque chaque transmission de trame sur le bus série implique l'interaction entre plusieurs modules fonctionnels, il y a deux façons pour décrire l'activité sur le bus série:

- 1) Le fonctionnement général du bus série peut être décrit pour chaque trame, notant les actions des différents types de modules dans chaque sous-trame (cela est l'approche utilisée dans ce chapitre).
- 2) Le fonctionnement de chaque module fonctionnel peut être décrit séparément, en tant qu'interaction entre le module et le bus série (cela est l'approche utilisée dans le chapitre 5).

CHAPTER 3: SERIAL BUS FRAMES AND SUBFRAMES

3.1 *Basic frame types*

As noted in Chapter 2, a frame is composed of subframes which are sent by several module groups in sequence whenever a Controller group sends a Header. Depending on the types of module groups that the Header selects, three basic frame types can occur on the serial bus, as shown in Figure 3-1, page 53.

A Cancelled frame indicates that a selected module group is not ready for the frame.

If all of the selected groups are ready, and the S address in the Header selects a Talker group, a Data Transfer frame results. Data Transfer frames send 1-32 bytes of data from Talker to Listener groups, and may also affect Flag groups that are used to "track" data transfers.

A Control frame results if all of the selected groups are ready and the S address has not selected a Talker group. Control frames do not contain any data, and are used only to set and reset Flag groups.

Table 3-1 defines which modules drive and receive the various subframes in each of these basic frame types.

Since each frame transmission on the serial bus involves interaction among several functional modules, there are two ways to describe the activity on the serial bus:

- 1) Overall serial bus operation can be described for each frame, noting the actions of the various module types in each subframe. (This is the approach used in this chapter.)
- 2) The operation of each functional module can be described separately, in terms of the interaction between it and the serial bus. (This is the approach used in chapter 5.)

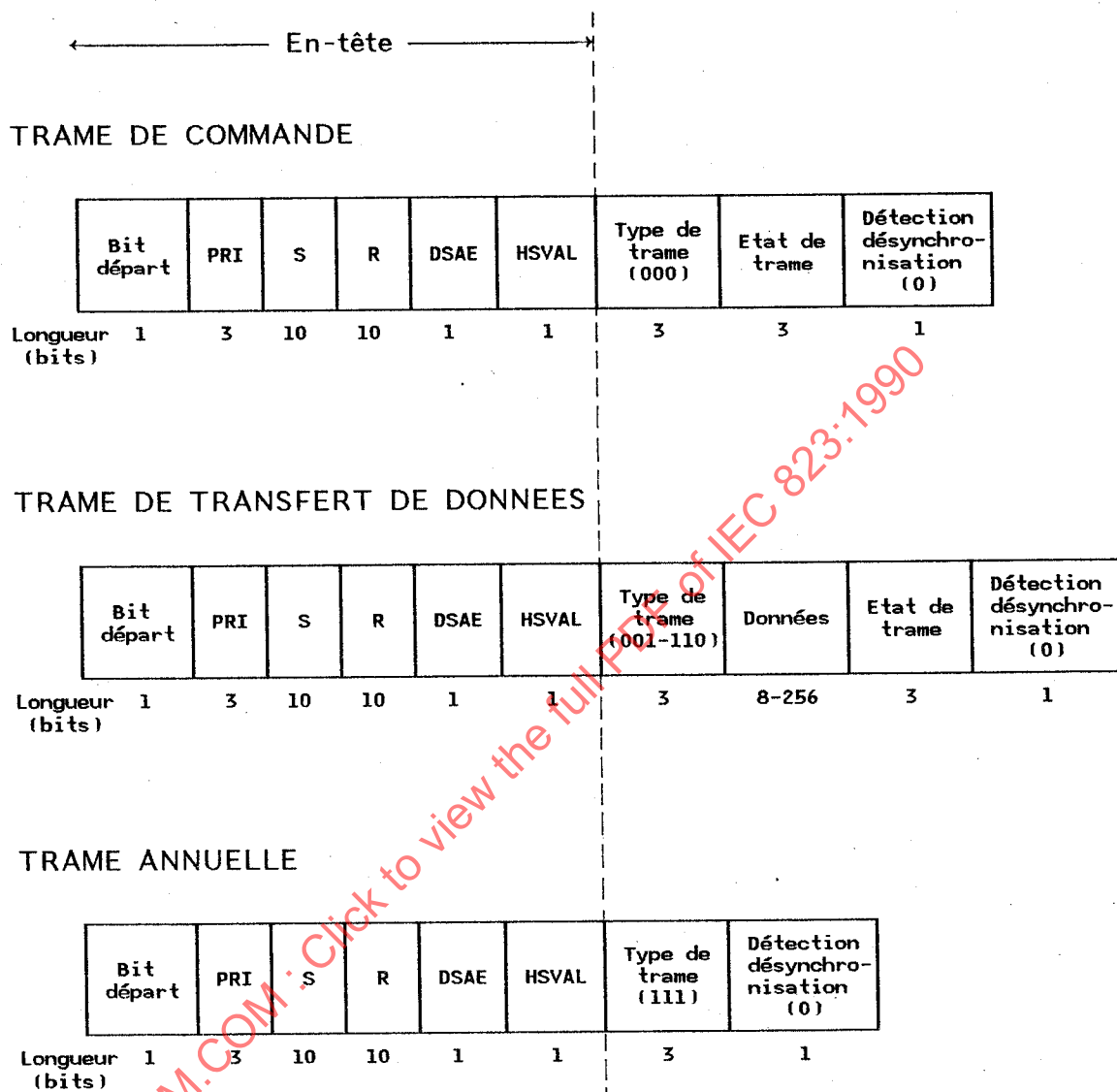


Figure 3-1 - Types généraux de trame

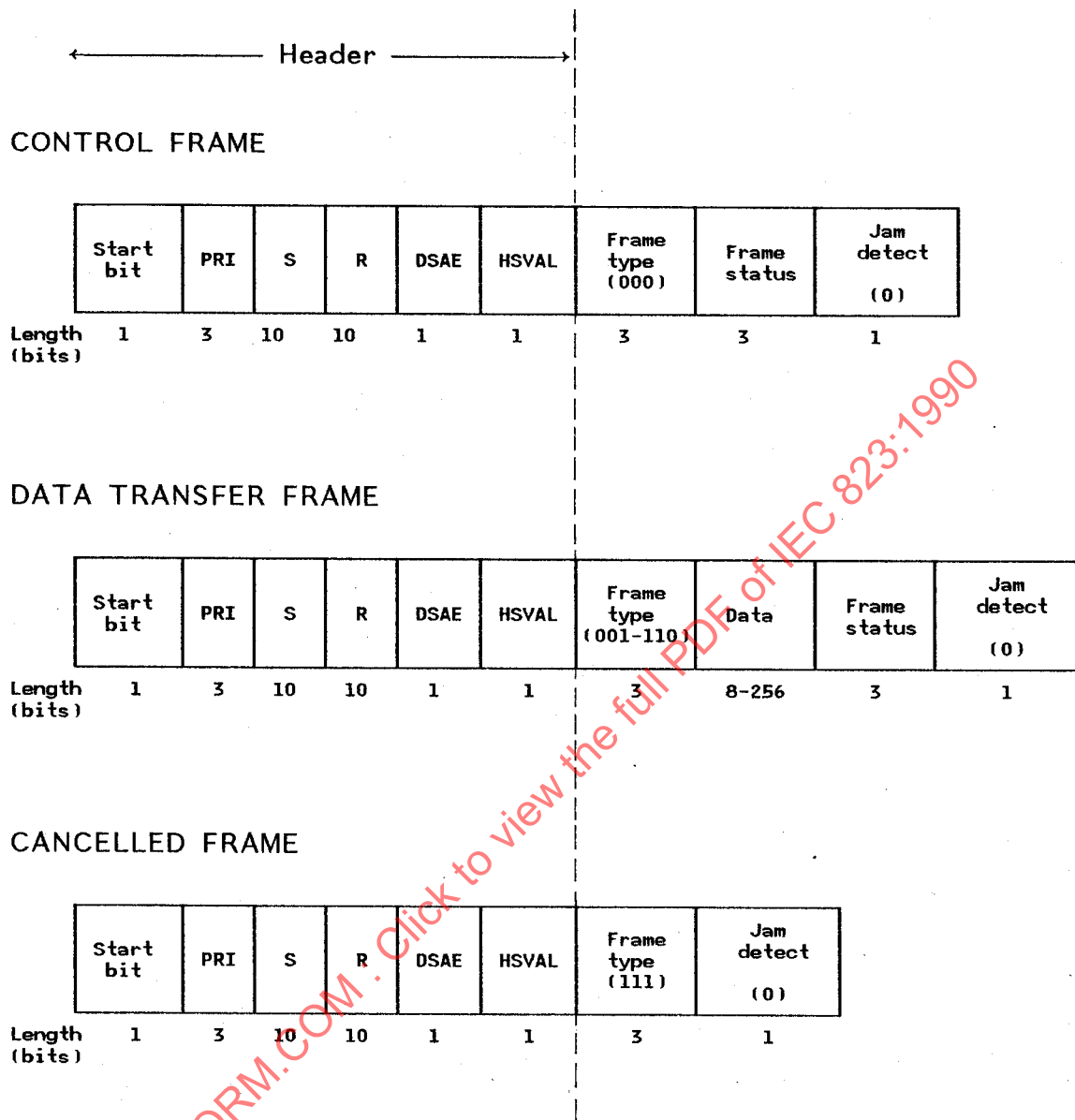


Figure 3-1 - Basic frame types

Tableau 3-1 - Utilisation des sous-trames par les modules

Trame	Sous-trame	Envoyée par le	Reçu par le
Commande	En-tête	EMETTEUR D'EN-TETE	RECEPTEUR D'EN-TETE, GESTIONNAIRE DE TRAME
	Type de trame	(Pas de module, par défaut à 000)	RECEPTEUR D'EN-TETE, GESTIONNAIRE DE TRAME
	Etat de la trame	RECEPTEUR D'EN-TETE	GESTIONNAIRE DE TRAME
	Bit de désynchronisation	(Pas de module, par défaut à 0)	RECEPTEUR D'EN-TETE, EMETTEUR DE DONNEES, RECEPTEUR DE DONNEES, GESTIONNAIRE DE TRAME
Transfert de données	En-tête	EMETTEUR D'EN-TETE	RECEPTEUR D'EN-TETE, GESTIONNAIRE DE TRAME
	Type de trame	EMETTEUR DE DONNEES (001-110)	RECEPTEUR DE DONNEES, GESTIONNAIRE DE TRAME
	Données	EMETTEUR DE DONNEES	RECEPTEUR DE DONNEES
	Etat de la trame	EMETTEUR DE DONNEES, RECEPTEUR DE DONNEES	GESTIONNAIRE DE TRAME, EMETTEUR DE DONNEES, RECEPTEUR DE DONNEES
	Bit de désynchronisation	(Pas de module, par défaut à 0)	RECEPTEUR D'EN-TETE, EMETTEUR DE DONNEES, RECEPTEUR DE DONNEES, GESTIONNAIRE DE TRAME
Annulée	En-tête	EMETTEUR D'EN-TETE	RECEPTEUR D'EN-TETE, GESTIONNAIRE DE TRAME
	Type de trame	RECEPTEUR D'EN-TETE (111)	GESTIONNAIRE DE TRAME, EMETTEUR DONNEES, RECEPTEUR DONNEES
	Bit de désynchronisation	(Pas de module, par défaut à 0)	RECEPTEUR D'EN-TETE, EMETTEUR DONNEES, RECEPTEUR DONNEES, GESTIONNAIRE DE TRAME

Toutes les trames commencent avec un en-tête et une sous-trame de type de trame et se terminent avec une sous-trame de détection de désynchronisation. L'en-tête détermine quels modules vont participer au reste de la trame, lesquels à leur tour vont déterminer le type de trame qui sera envoyée. La sous-trame de type de trame fait connaître le type de la trame à tous les modules du bus série et spécifie la longueur de la trame. La sous-trame de détection de désynchronisation "valide" la trame complète et assure la "synchronisation de la trame" entre tous les modules du bus série.

Table 3-1 - Subframe usage by modules

Frame	Subframe	Sent by	Received by
Control	Header	HEADER SENDER	HEADER RECEIVER, FRAME MONITOR
	Frame type	(No module, defaults to 000)	HEADER RECEIVER, FRAME MONITOR
	Frame status	HEADER RECEIVER	FRAME MONITOR
	Jam bit	(No module, defaults to 0)	HEADER RECEIVER, DATA SENDER, DATA RECEIVER, FRAME MONITOR
Data transfer	Header	HEADER SENDER	HEADER RECEIVER, FRAME MONITOR
	Frame type	DATA SENDER (001-110)	DATA RECEIVER, FRAME MONITOR
	Data	DATA SENDER	DATA RECEIVER
	Frame status	DATA SENDER, DATA RECEIVER	FRAME MONITOR, DATA SENDER, DATA RECEIVER
	Jam bit	(No module, defaults to 0)	HEADER RECEIVER, DATA SENDER, DATA RECEIVER, FRAME MONITOR
Cancelled	Header	HEADER SENDER	HEADER RECEIVER, FRAME MONITOR
	Frame type	HEADER RECEIVER (111)	FRAME MONITOR, DATA SENDER, DATA RECEIVER
	Jam bit	(No module, defaults to 0)	HEADER RECEIVER, DATA SENDER, DATA RECEIVER, FRAME MONITOR

All frames start with a Header and a Frame Type subframe, and end with a Jam Detect subframe. The Header determines which modules will participate in the rest of the frame, which in turn determines the type of frame that is sent. The Frame Type subframe makes the type of frame known to all serial bus modules, and specifies the length of the frame. The Jam Detect subframe "validates" the complete frame and ensures "frame synchronization" among all serial bus modules.

3.2 En-tête

Un en-tête est composé de 26 bits organisés en six champs:

Bit départ	Priorité de trame	S	R	EMETTEUR DE DONNEES Validation arbitrage	HSVAL
1	3	10	10	1	1

Le contenu de chacun de ces champs est le suivant:

- 1) Un bit_départ unique. Lorsque le bit_départ est transmis, il signale aux modules du bus série que la trame commence.
- 2) Un champ de trois bits "priorité de la trame". Si deux ou plusieurs groupes parleurs essaient d'envoyer une trame en même temps, ce champ est utilisé pour arbitrer le contrôle du bus série afin que la trame de priorité la plus élevée soit envoyée en premier.
- 3) Une adresse S de dix bits, qui peut sélectionner un ou plusieurs groupes de modules du bus série. Cette adresse peut sélectionner des indicateurs qui sont à positionner et/ou des parleurs qui vont envoyer des données dans la sous-trame de données qui suit. (L'adresse S peut sélectionner plusieurs groupes si leurs RECEPTEURS EN-TETE ont tous été configurés pour la reconnaître.)
- 4) Une adresse R de dix bits, qui peut sélectionner un ou plusieurs groupes de modules du bus série. Cette adresse peut sélectionner des indicateurs qui sont à remettre à zéro et/ou des écouteurs qui vont capter les données à partir de la sous-trame de données qui suit. Dans une trame prévue pour positionner un groupe sémaphore, cette adresse est un "code demandeur" qui ne sélectionne pas un groupe, mais garantit qu'un seul groupe gestionnaire gagne l'arbitrage du bus série dans l'en-tête. (L'adresse R peut sélectionner plusieurs groupes si leurs RECEPTEURS EN-TETE ont été tous initialisés pour la reconnaître.)

OBSERVATION 3.1:

Le chapitre 6 donne des informations ultérieures concernant l'utilisation de l'adresse R pour les "codes du demandeur".

Une trame peut ne pas avoir d'autre but que de positionner ou de remettre à zéro un indicateur, sans exiger que ce soit effectué par un gestionnaire unique. Dans un tel cas, une des deux adresses de l'en-tête est effectivement "non utilisée". Le bus série réserve la valeur "tous à un" (décimal 1023) pour l'utiliser comme champ d'adresse "non utilisé".

3.2 The Header

A Header is composed of 26 bits arranged in six fields:

Start bit	Frame priority	S	R	DATA SENDER Arb enable	HSVAL
1	3	10	10	1	1

The contents of each of these fields are as follows:

- 1) A single start_bit. When the start_bit is transmitted, it signals serial bus modules that a frame is beginning.
- 2) A three-bit "Frame Priority" field. If two or more Talker groups attempt to send a frame at the same time, this field is used to arbitrate control of the serial bus so that the frame with higher priority is sent first.
- 3) A ten-bit S address, which may select one or more serial bus module group(s). This address may select Flags that are to be set, and/or Talkers that are to send data in the Data subframe which follows. (The S address may select several groups if their HEADER RECEIVERS have all been configured to recognize it.)
- 4) A ten-bit R address, which may select one or more serial bus module group(s). This address may select Flags that are to be reset, and/or Listeners that are to capture the data in the Data subframe which follows. In a frame intended to set a Semaphore group, this address is a "Requester code" which does not select a group, but rather guarantees that only one Controller group wins the serial bus arbitration in the Header. (The R address may select several groups if their HEADER RECEIVERS have all been set up to recognize it.)

OBSERVATION 3.1:

Chapter 6 gives further information about the use of the R address for "Requester codes".

A frame might have no other purpose than to set or reset a Flag, with no requirement that this be done by a unique Controller. In such a case, one of the two addresses in the Header is effectively "unused". The serial bus reserves the "all ones" value (decimal 1023) for use in an "unused" address field.

REGLE 3.1:

Les couches supérieures de gestion du bus série NE DOIVENT configurer aucun groupe pour répondre à l'adresse "tous à un" (décimal 1023).

OBSERVATION 3.2:

Cette norme ne définit pas le fonctionnement des groupes de modules du bus série lorsqu'ils sont sélectionnés par les deux adresses S et R dans l'en-tête.

RECOMMANDATION 3.1:

En dehors des besoins de diagnostic, ne pas envoyer des en-têtes avec la même adresse dans les champs S et R. De même, ne pas envoyer des en-têtes avec des adresses différentes S et R, mais lesquelles, par l'adressage de groupe, vont sélectionner le même groupe de modules.

- 5) Un bit de validation d'arbitrage de l'EMETTEUR DE DONNEES (DSAE). Lorsque deux ou plusieurs parleurs partagent une adresse commune, ils seront sélectionnés pour émettre en même temps. Dans ce cas, la sous-trame de données qui en résulte peut contenir soit le OU logique de leurs valeurs, soit la plus grande valeur parmi les données des parleurs. Pour spécifier que le OU logique est demandé, le gestionnaire envoie un bit_zéro dans cette position de bit. Pour spécifier que la valeur la plus grande est demandée, il envoie un bit_un. (Un bit_un commande à chaque parleur d'utiliser l'arbitrage du bus série lorsqu'il envoie les données, c'est-à-dire arrêter d'envoyer les données s'il échantillonne un bit_un dans une position de bit où il a envoyé un bit_zéro. Un bit_zéro indique aux parleurs d'omettre l'arbitrage et de maintenir l'émission sans tenir compte des données échantillonnées.)

OBSERVATION 3.3:

Le bit DSAE n'a pas de fonction définie dans une trame de commande.

- 6) Un bit de validation de l'EMETTEUR D'EN-TETE (HSVAL). Ce bit est envoyé toujours comme un bit_un par un gestionnaire à la fin de l'en-tête. Ce bit est spécifié comme faisant partie de l'en-tête, pour assurer qu'une erreur quelconque sur le bus série ne va pas permettre à tous les gestionnaires de croire qu'ils ont perdu l'arbitrage et de se retirer du bus. Si cela arrive, le gestionnaire insère une chaîne de bit_zéro dans le reste de l'en-tête. Cela pourrait être interprété par erreur par d'autres modules comme un en-tête valide.

Le bit de validation résout ce problème parce que si tous les parleurs se retirent, ce bit sera un bit_zéro, faisant ainsi que tous les modules du bus série ignorent l'en-tête.

RULE 3.1:

Higher layers of serial bus management **MUST NOT** configure any serial bus group to respond to the "all ones" address (decimal 1023).

OBSERVATION 3.2:

This standard does not define the operation of serial bus module groups when they are selected by both the S and R addresses in a Header.

RECOMMENDATION 3.1:

Except for diagnostic purposes, do not send Headers with the same address in the S and R fields. Similarly, do not send Headers with S and R addresses that are different but which, due to group addressing, both select the same module group.

- 5) A DATA SENDER arbitration enable bit (DSAE). When two or more Talkers share a common address, they will be selected to send at the same time. In this case, the resulting Data subframe can contain either the logical OR of their values, or the largest value among the Talkers' data. To specify that a logical OR is required, the Controller sends a zero_bit in this bit position. To specify that the largest value is required, it sends a one_bit. (A one_bit commands each Talker to use serial bus arbitration while sending the data, that is, to stop sending data if it samples a one_bit in a bit in which it sent a zero_bit. A zero_bit tells the Talkers to omit arbitration, and keep sending regardless of the sampled data.)

OBSERVATION 3.3:

The DSAE bit has no defined function in a Control frame.

- 6) A HEADER SENDER validation bit (HSVAL). This bit is always sent as a one_bit by a Controller at the conclusion of the Header. This bit is specified as part of the Header, to ensure that some error on the serial bus would not cause all the Controllers to think they have lost the arbitration and retire from the bus. If this occurs, the Controller leaves a string of zero_bits in the rest of the Header. This could be misinterpreted by other modules as a valid Header.

The validation bit solves this problem because if all of the Talkers retire it will be a zero_bit, causing all serial bus modules to ignore the Header.

3.3 Sous-trame de type de trame

La deuxième sous-trame présente dans toutes les trames est la sous-trame de type de trame. Elle est composée de trois bits:

Bit le plus
significatif (MSB)

bit 2	bit 1	bit 0
-------	-------	-------

Le code de ces bits différencie les trois types de base de trames du bus série: trames de commande, trames de transfert de données et trames annulées. Le tableau 3-2 montre la signification des différents codes de types de trame.

Lorsqu'un parleur est sélectionné par l'adresse S, il envoie un code de type de trame de trois bits dans cette sous-trame. Ce code indique combien d'octets de données le parleur a l'intention d'envoyer.

Lorsqu'un parleur, écouteur ou un groupe indicateur est sélectionné par l'adresse S ou R, il peut annuler la trame en envoyant 111 dans cette sous-trame, s'il reçoit l'indication de faire ainsi par une couche supérieure de gestion du bus série:

Tableau 3-2 Codes de type de trame

Code	Signification
000	Trame de commande
001	Trame de transfert de données de 1 octet
010	Trame de transfert de données de 2 octets
011	Trame de transfert de données de 4 octets
100	Trame de transfert de données de 8 octets
101	Trame de transfert de données de 16 octets
110	Trame de transfert de données de 32 octets
111	Trame annulée

Puisque les données sont en logique OU sur le bus série, l'annulation (111) domine les codes de longueurs (001-110) envoyés par les parleurs. Si aucun parleur, écouteur ou indicateur n'annule la trame, et qu'aucun parleur n'envoie un code de longueur, une trame de commande (000) en résulte.

3.3 The Frame Type subframe

The second subframe present in all frames is the Frame Type subframe. It is composed of three bits:

Most significant bit (MSB)	bit 2	bit 1	bit 0
-------------------------------	-------	-------	-------

The code in these bits differentiates the three basic types of serial bus frames: Control frames, Data Transfer frames and Cancelled frames. Table 3-2 shows the meaning of the various Frame Type codes.

When a Talker is selected by the S address, it sends a three bit Frame Type code in this subframe. This code indicates how many bytes of data the Talker is planning to send.

When a Talker, Listener or Flag group is selected by either the S or the R address, it may cancel the frame by sending 111 in this subframe, if it is signalled to do so by a higher layer of serial bus management.

Table 3-2 - Frame Type codes

Code	Meaning
000	Control frame
001	1 byte data transfer frame
010	2 byte data transfer frame
011	4 byte data transfer frame
100	8 byte data transfer frame
101	16 byte data transfer frame
110	32 byte data transfer frame
111	Cancelled frame

Since data is OR'ed on the serial bus, a Cancellation (111) predominates over the length code (001-110) sent by Talkers. If no Talker, Listener or Flag cancels the frame, and no Talker sends a length code, a Control frame (000) results.

En général, l'annulation d'une trame indique qu'un groupe n'est "pas prêt" pour participer à la trame. Cette norme ne définit pas toutes les conditions locales sur la carte qui peuvent être la cause d'annulation de la trame par un groupe, mais les conditions les plus fréquentes sont les suivantes:

- 1) Si un groupe écouteur a reçu des données dans la trame précédente et ne les a pas encore "utilisées". Afin d'éviter la perte de données de la trame précédente, le récepteur de données du groupe signale à son RECEPTEUR D'EN-TETE qu'il n'est pas disponible pour une autre trame. Si le gestionnaire envoie un autre en-tête avec l'adresse de l'écouteur dans le champ R, avant d'avoir utilisé les données précédentes, le RECEPTEUR D'EN-TETE annule la nouvelle trame en envoyant 111 dans le type de trame. Les parleurs échantillonnent la sous-trame de type de trame dès qu'ils l'envoient et arrêtent l'émission s'ils échantillonnent 111. (Une trame annulée se termine après les bits de type de trame.)
- 2) Si un groupe parleur n'est pas alimenté en données par les couches supérieures de gestion du bus série. Comme pour le groupe écouteur du cas précédent, l'EMETTEUR DE DONNEES du parleur le signale à son RECEPTEUR D'EN-TETE, par un signal sur la carte. Si un gestionnaire envoie une trame avec l'adresse du parleur dans le champ S alors que le parleur "n'a pas de données", son RECEPTEUR D'EN-TETE annule la trame en envoyant 111 dans la sous-trame de type de trame. Les écouteurs sélectionnés échantillonnent ce code et ignorent la trame.

Un système peut être organisé de manière que plusieurs parleurs soient sélectionnés par une adresse particulière S. Si des parleurs multiples sont sélectionnés par l'adresse S et un ou plusieurs d'entre eux ne sont pas prêts à émettre, ceux qui sont prêts échantillonnent le code 111 et n'envoient aucune donnée.
- 3) Un groupe sémaphore ou à passage de jeton peut également envoyer 111 dans la sous-trame de type de trame afin d'annuler la trame. S'il est sélectionné par l'adresse S, et que sa bascule est déjà positionnée, ou (dans un groupe à passage de jeton) s'il est sélectionné par l'adresse R et que sa bascule est déjà remise à zéro, alors il annule la trame.

OBSERVATION 3.4:

Comme décrit dans les chapitres 5 et 6, les groupes parleur et écouteur ne sont pas obligés d'annuler les trames, de la façon expliquée plus haut. Certains types de parleurs et d'écouteurs sont toujours prêts pour une trame.

En résumé, le RECEPTEUR D'EN-TETE dans un groupe parleur, écouteur ou indicateur envoie 111 pour annuler une trame s'il est sélectionné par l'adresse S ou R, et que son EMETTEUR DE DONNEES, RECEPTEUR DE DONNEES, ou autre circuit logique sur la carte a signalé qu'il devrait annuler une telle trame.

Quand une trame est annulée, ce fait est enregistré par le groupe gestionnaire qui a déclenché la trame, et ce fait est signalé aux couches supérieures de la gestion du bus série.

In general, cancelling a frame indicates that a group is "not ready" for the frame. This standard does not define all of the local/on-board conditions which may cause a group to cancel a frame, but the most common conditions are as follows:

- 1) If a Listener group has received data in a previous frame and has not "disposed" of that data yet. In order to keep from losing the data from the previous frame, the group's DATA RECEIVER signals its HEADER RECEIVER that it is not ready for another frame. If a Controller sends another Header with the Listener's address in the R field, before the previous data has been disposed of, the HEADER RECEIVER cancels the new frame by sending 111 in the Frame Type. Talkers sample the Frame Type subframe as they send it, and stop sending if they sample 111. (A Cancelled frame ends after the Frame Type bits.)
- 2) If a Talker group has not been provided with data by higher layers of serial bus management. As in the Listener group in the previous case, the Talker's DATA SENDER signals its HEADER RECEIVER of this fact via an on-board signal. If a Controller sends a frame with the Talker's address in the S field while the Talker "has no data", its HEADER RECEIVER cancels the frame by sending 111 in the Frame Type subframe. Selected Listeners sample this code and ignore the frame.

A system can be set up so that more than one Talker is selected by a particular S address. If multiple Talkers are selected by the S address, and one or more of them is not ready to send, those which are ready sample the 111 code and do not send any data.

- 3) A semaphore or token passing group may also send 111 in the Frame Type subframe to cancel a frame. If it is selected by the S address and its latch is already set, or (in a token passing group) if it is selected by the R address and its latch is already reset, it cancels the frame.

OBSERVATION 3.4:

As described in Chapters 5 and 6, Talker and Listener groups are not required to cancel frames as described above. Some types of Talkers and Listeners are always ready for a frame.

In summary, the HEADER RECEIVER in a Talker, Listener or Flag group sends 111 to cancel a frame if it is selected by either the S or the R address, and its DATA SENDER, DATA RECEIVER, or other on-board logic has signalled that it should cancel such a frame.

When a frame is cancelled, this fact is captured by the Controller group that initiated the frame, and is reported to higher layers of serial bus management.

3.4 Sous-trame de données

Cette sous-trame apparaît uniquement dans les trames de transfert de données. Sa longueur est spécifiée par le ou les parleurs sélectionnés dans la sous-trame de type de trame. Il peut y avoir 1, 2, 4, 8, 16 ou 32 octets dans une sous-trame de données. Comme décrit auparavant, il peut y avoir un ou plusieurs parleurs émettant dans cette sous-trame, selon le nombre de parleurs qui ont été configurés pour répondre à l'adresse S. Si plusieurs parleurs sont sélectionnés, la sous-trame de données contiendra soit le OU logique de leurs données soit la plus grande des valeurs (cela est déterminé par le fait que la logique d'arbitrage du parleur était invalidée ou non par le bit de validation d'arbitrage de l'émetteur de données envoyé dans l'en-tête).

3.5 Sous-trame d'état de trame

Tandis que l'annulation d'une trame peut être déterminée à partir du champ de type de trame, d'autres problèmes peuvent se présenter quand une trame est envoyée. Comme les annulations, de tels problèmes sont signalés aux couches supérieures de la gestion du bus série, par le groupe gestionnaire qui a déclenché la trame. La sous-trame d'état de trame est incluse à la fin de chaque trame, excepté pour les trames annulées et est utilisée pour signaler si les groupes de modules ont été sélectionnés par les adresses S ou R. Elle est également utilisée pour signaler des problèmes (des conditions exceptionnelles) qui auraient pu survenir pendant l'émission de la trame. La sous-trame d'état de trame est composée de trois bits:

Bit le plus
significatif (MSB)

bit 2	bit 1	bit 0
-------	-------	-------

Toutes les fois qu'un RECEPTEUR D'EN-TETE reçoit un en-tête avec une adresse S ou R qui correspond à la sienne, il positionne une des deux sorties sur la carte à l'état vrai. Selon que le RECEPTEUR D'EN-TETE fait partie d'un groupe parleur, écouteur ou indicateur, ces sorties peuvent ou non valider un EMETTEUR DE DONNEES ou RECEPTEUR DE DONNEES sur la carte.

Par la suite, le RECEPTEUR D'EN-TETE échantillonne la sous-trame de type de trame. Si un groupe parleur est sélectionné, son EMETTEUR DE DONNEES envoie une des valeurs entre 001 et 110 dans ce type de trame. Si le RECEPTEUR D'EN-TETE échantillonne 001-110, il ignore la sous-trame de données et il échantillonne ces trois bits d'état de trame et le bit suivant de détection de désynchronisation. (Dans ce cas, le RECEPTEUR D'EN-TETE joue simplement le rôle de "traqueur" afin d'indiquer comment les EMETTEURS DE DONNEES et RECEPTEUR DE DONNEES répondent à la trame.) Si l'état de la trame

3.4 The Data subframe

This subframe occurs only in Data Transfer frames. Its length is specified by the selected Talker(s) in the Frame Type subframe. There may be 1, 2, 4, 8, 16 or 32 bytes in a Data subframe. As described earlier, there may be one or more Talker(s) sending in this subframe, depending on how many are configured to respond to the S address. If several Talkers are selected, the Data subframe will either contain the logical OR of their data, or the largest value among them. (This is determined by whether the Talkers' arbitration logic was disabled or enabled by the Data Sender Arbitration Enable bit sent in the Header.)

3.5 The Frame Status subframe

While the cancellation of a frame can be determined from the Frame Type field, other problems can arise when a frame is sent. Like cancellations, such problems are reported to higher layers of serial bus management, by the Controller group that initiated the frame. The Frame Status subframe is included at the end of every frame except a Cancelled frame, and is used to signal whether module groups were selected by the S and R addresses. It is also used to signal any problems (exceptional conditions) that might have arisen during the frame's transmission. The Frame Status subframe is composed of three bits:

Most significant
bit (MSB)

bit 2	bit 1	bit 0
-------	-------	-------

Whenever a HEADER RECEIVER receives a Header with an S or an R address that matches its own, it makes one of two on-board outputs True. Depending on whether the HEADER RECEIVER is part of a Talker, Listener or Flag group, these outputs may or may not enable an on-board DATA SENDER or DATA RECEIVER.

The HEADER RECEIVER then samples the Frame Type subframe. If a Talker group is selected, its DATA SENDER sends one of the values 001 through 110 in the Frame Type. If the HEADER RECEIVER samples 001-110, it passes over the Data subframe and samples these three Frame Status bits and the following Jam Detect bit. (In this case the HEADER RECEIVER is simply acting as a "tracker" to reflect how DATA SENDERS and DATA RECEIVERS responded to the frame.) If

est 011 et le bit de détection de désynchronisation est un bit_zéro, le RECEPTEUR D'EN-TETE émet des impulsions sur l'une des deux sorties de la carte. Selon le type de groupe de modules dans lequel se trouve le RECEPTEUR D'EN-TETE, cette sortie peut positionner ou remettre à zéro une bascule "de poursuite" sur la carte.

Si le RECEPTEUR D'EN-TETE échantillonne 000 dans le type de trame, cela indique qu'aucun parleur n'a été sélectionné par la trame. Le RECEPTEUR D'EN-TETE échantillonne le bit le plus significatif du champ d'état de trame (immédiatement derrière) et envoie ensuite un 10 ou 01 dans les deux bits suivants pour indiquer qu'il a été sélectionné par l'adresse S ou R, respectivement. Puis il échantillonne le bit de détection de désynchronisation. Si le bit le plus significatif de l'état de trame et le bit de détection de désynchronisation sont tous les deux des bit_zéro, la trame de contrôle est considérée comme réussie et le RECEPTEUR D'EN-TETE envoie des impulsions sur l'une des deux sorties afin de positionner ou remettre à zéro une bascule sur la carte.

Si deux parleurs ou plus sont sélectionnés par l'adresse S dans l'en-tête, le protocole du bus série exige qu'ils soient en accord sur la quantité de données à envoyer. Pendant que chaque parleur envoie son code dans le champ de type de trame pour signaler combien de données il doit envoyer, il échantillonne aussi ces bits afin de détecter un conflit avec d'autres parleurs. Si un parleur échantillonne un bit_un dans un des trois bits, pendant qu'il envoie un bit_zéro, il arrête d'envoyer davantage de bits dans la sous-trame de type de trame ou dans la sous-trame de données. Pourtant, dans ce cas, il n'échantillonne pas tous les bits dans le type de trame. Si la valeur échantillonnée n'est pas 111 (pas d'annulation), le parleur utilise la valeur échantillonnée pour déterminer la longueur de la sous-trame de données, ignore les données et, par la suite, envoie 110 pendant la sous-trame d'état de trame pour indiquer un conflit entre parleurs sélectionnés concernant la taille de la sous-trame de données.

Si un parleur échantillonne la même valeur dans le type de trame qu'il envoie, il envoie ses données. Puis il échantillonne le bit le plus significatif de la trame d'état, qui indique un problème de taille de données entre d'autres parleurs et écouteurs. Si le bit le plus significatif est un bit_zéro, le parleur envoie 10 dans les deux bits suivants de l'état de la trame pour indiquer qu'il a survécu à l'arbitrage dans la sous-trame de données.

Supposons qu'un parleur soit sélectionné par l'adresse S dans l'en-tête et qu'un écouteur soit sélectionné par l'adresse R. Supposons, de plus, que l'écouteur soit seulement capable de recevoir des sous-trames de données jusqu'à une certaine longueur (moins de 32 octets). L'écouteur échantillonne la sous-trame de type de trame. Si le code de la sous-trame indique que le parleur essaie d'envoyer plus de données qu'il ne peut en traiter, l'écouteur compte les bits à travers la taille des données indiquée par le type de trame et envoie ensuite 101 dans la sous-trame d'état de trame pour signaler le problème.

Si un écouteur sélectionné échantillonne 000 dans le champ du type de trame, il n'y a pas de parleur sélectionné pour lui envoyer des données. Il traite cette situation en émettant 101 dans la sous-trame d'état de la trame.

the Frame Status is 011 and the Jam Detect bit is a zero_bit, the HEADER RECEIVER pulses one of two on-board outputs. Depending on what type of module group the HEADER RECEIVER is in, this output may set or clear an on-board "tracking" latch.

If the HEADER RECEIVER samples 000 in the Frame Type, this indicates that no Talker was selected by the frame. The HEADER RECEIVER samples the MSB of the (immediately following) Frame Status field, and then sends either a 10 or a 01 in the next 2 bits to indicate that it was selected by the S or the R address, respectively. It then samples the Jam Detect bit. If the MSB of the Frame Status and the Jam Detect bits are both zero_bits, the Control frame is considered successful, and the HEADER RECEIVER pulses one of two outputs to set or clear an on-board latch.

If two or more Talkers are selected by the S address in the Header, the serial bus protocol requires that they agree about how much data is to be sent. As each Talker sends its code in the Frame Type field to signal how much data it has to send, it also samples these bits to detect a conflict with other Talkers. If a Talker samples a one_bit in any of the 3 bits while it is sending a zero_bit, it stops sending any further bits in the Frame Type subframe or Data subframe. However, in this case it does sample all the bits in the Frame Type. If the sampled value is not 111 (i.e. not a cancellation), the Talker uses the sampled value to determine the length of the Data subframe, passes over the data, and thereafter sends 110 during the Frame Status subframe to signify a conflict among selected Talkers as to the size of the Data subframe.

If a Talker samples the same value in the Frame Type that it is sending, it sends its data. It then samples the MSB of the Frame Status, which indicates a data-size problem between other Talkers and Listeners. If the MSB is a zero_bit, the Talker sends 10 in the next two bits of the Frame Status to indicate that it survived any arbitration in the Data subframe.

Suppose that a Talker is selected by the S address in the Header, and a Listener is selected by the R address. Let us further suppose that the Listener is only capable of receiving Data subframes up to a certain length (less than 32 bytes). The Listener samples the Frame Type subframe. If the Frame Type code indicates that the Talker is trying to send more data than it can handle, the Listener counts bits across the size of data indicated by the Frame Type, and then sends 101 in the Frame Status subframe to signal the problem.

If a selected Listener samples 000 in the Frame Type field, there is no Talker selected to send data to it. It handles this situation by sending 101 in the Frame Status subframe.

Autrement (pour les types de trames 001-110), l'écouteur échantillonne le nombre d'octets de données indiqué par le type de trame. Puis, il échantillonne le bit le plus significatif de la sous-trame d'état de la trame. Si le bit le plus significatif est un bit_zéro, il envoie 01 dans les deux bits suivants de l'état de trame pour indiquer qu'il a été sélectionné. Ensuite, il échantillonne le bit de détection désynchronisation. Si le bit le plus significatif de la trame d'état et le bit de détection de désynchronisation étaient tous les deux des bit_zéro, il présente les données qu'il a échantillonnées aux couches supérieures de gestion du bus série, avec une sortie indiquant aux couches supérieures l'arrivée de données.

En résumé, la valeur de l'état de trame est interprétée de façon différente pour une trame de transfert de données et pour une trame de commande. Dans chaque cas, il y a huit valeurs d'état possibles, comme indiqué dans le tableau 3-3.

3.6 Sous-trame de détection de désynchronisation

Cette "sous-trame" consiste en un bit unique. Elle peut être considérée comme une sous-trame ou comme "le bit après une trame". Aucun des modules qui envoient d'autres sous-trames ne positionne jamais ce bit comme un bit_un mais les parleurs, écouteurs et indicateurs l'échantillonnent tous pour s'assurer que c'est un bit_zéro. Si jamais le bit de détection de désynchronisation est échantillonné par ces modules comme différent d'un bit_zéro, ils ignorent la trame d'émission précédente.

3.7 Protocole de resynchronisation

La procédure décrite ci-dessus est nécessaire car, en présence d'un bruit élevé sur le bus série, les gestionnaires peuvent se désynchroniser. C'est-à-dire qu'ils peuvent perdre les repères de début et de fin de trames. Cette situation doit être détectée et les modules resynchronisés avant que l'information des trames ne soit déformée et mal interprétée. Si les gestionnaires se désynchronisent, l'un d'eux commence à émettre une trame pendant la transmission d'une autre trame.

Les GESTIONNAIRES DE TRAME dans les groupes gestionnaires vérifient constamment les bit_départ. Si un GESTIONNAIRE DE TRAME détecte un bit_départ alors qu'aucune trame n'est en cours, il suit la transmission jusqu'à la fin de la trame. S'il échantillonne un autre bit_départ avant qu'il ait terminé le comptage de la trame, alors il "invalidé" la transmission de trames en émettant une séquence de 512 bit_un ou un bit_départ suivi par 511 bit_un.

Ainsi, une trame "invalidée" peut ressembler à n'importe quel type de trame de base indiquée dans la figure 3-1, page 52. Elle est différente par le fait (au moins) que le contenu du bit de détection de désynchronisation est différent du bit_zéro de la figure 3-1.

Puisque toutes les trames du bus série ont moins de 512 bits, la séquence de resynchronisation garantit qu'aucun module du bus série ne reçoit un bit_zéro dans la position du bit de détection de désynchronisation de la trame courante, quelle que soit sa façon de suivre les trames.

Otherwise (for Frame Types 001-110) the Listener samples the number of bytes of data indicated by the Frame Type. It then samples the MSB of the Frame Status subframe. If the MSB is a zero_bit, it sends 01 in the next two bits of the Frame Status to show that it was selected. It then samples the Jam Detect bit. If the MSB of the Frame Status and the Jam Detect bit were both zero_bits, it presents the data it sampled to higher layers of serial bus management, along with an output that indicates the arrival of data to higher layers.

In summary, the Frame Status value is interpreted differently for a Data Transfer frame versus a Control frame. In each, case there are eight possible status values, as shown in Table 3-3.

3.6 *The Jam Detect subframe*

This "subframe" consists of a single bit. It can be regarded as a subframe or as "the bit after a frame". None of the modules which send other subframes ever drives this bit as a one_bit, but Talkers, Listeners and Flags all sample it to be sure it is a zero_bit. If the Jam Detect bit is ever sampled by these modules as other than a zero_bit, they ignore the preceding frame transmission.

3.7 *Jam protocol*

The procedure described above is necessary because, if a high level of noise is induced onto the serial bus, Controllers may become desynchronized. That is, they may lose track of where frames begin and end. This condition has to be detected, and the modules resynchronized, before frame information is distorted and misinterpreted. If Controllers become desynchronized, one of them begins to send a frame during the transmission of another frame.

The FRAME MONITORS in Controller groups constantly check for start_bits. When a FRAME MONITOR detects a start_bit when *no* frame is in progress, it tracks the transmission to the end of the frame. If it samples another start_bit before it has finished counting out the frame, then it "jams" the frame transmission by sending a sequence of 512 one_bits, or a Start_bit followed by 511 one_bits.

Thus, a "jammed" frame may look like any of the basic frame types shown in Figure 3-1, page 53. It differs from them in that (at least) the Jam Detect bit contains something other than the zero_bit shown in Figure 3-1.

Since all serial bus frames are shorter than 512 bits, the Jam Sequence ensures that no serial bus module receives a zero_bit in the Jam Detect bit position of the current frame, regardless of how it may be tracking frames.

Tableau 3-3 - Codes de la trame d'état

Trame	Valeur de l'état	Signification
Contrôle	000	Pas de groupe de modules sélectionné par S ou R
	001	Groupe indicateur sélectionné par R seulement
	010	Indicateur sélectionné par S seulement
	011	Indicateurs sélectionnés par S et R
	100	RESERVE: ne devrait pas avoir lieu
	101	Ecouteur sélectionné par R, rien sélectionné par S
	110	RESERVE: ne devrait pas avoir lieu
	111	Ecouteur sélectionné par R, indicateur (seulement) sélectionné par S
Transfert de données	000	Pas d'écouteurs sélectionnés par R, tous les parleurs sélectionnés par S ont perdu l'arbitrage du champ de données
	001	Ecouteur sélectionné par R, tous les parleurs sélectionnés par S ont perdu l'arbitrage du champ de données
	010	Aucun écouteur sélectionné par R, le ou les parleurs sélectionnés par S ont terminé normalement la sous-trame de données
	011	Transfert de données terminé correctement
	100	RESERVE: ne devrait pas avoir lieu
	101	Ecouteur(s) incapable(s) de traiter la taille des données
	110	Tailles des données incompatibles entre les parleurs
	111	(Les deux conditions 101 et 110)

Parce que la trame en cours a été endommagée par un bit_départ mal placé (et typiquement aussi par la séquence de resynchronisation), une trame contenant autre chose qu'un bit_zéro dans le bit de détection de désynchronisation, est ignorée par tous les modules du bus série. Dans ce cas, aucun indicateur n'est positionné ou remis à zéro, et aucun parleur ni écouteur ne considère qu'il a envoyé ou reçu des données.

Table 3-3 - Frame Status codes

Frame	Status value	Meaning
Control	000	No module group selected by either S or R
	001	Flag group selected by R only
	010	Flag selected by S only
	011	Flag selected by S and R
	100	RESERVED: should not occur
	101	Listener selected by R, nothing selected by S
	110	RESERVED: should not occur
	111	Listener selected by R, Flag (only) selected by S
Data transfer	000	No Listener selected by R, all Talkers selected by S have lost Data field arbitration
	001	Listener selected by R, all Talkers selected by S have lost Data field arbitration
	010	No Listener selected by R, Talker(s) selected by S have completed the Data subframe normally
	011	Data Transfer completed correctly
	100	RESERVED: should not occur
	101	Listener(s) unable to handle data size
	110	Data size conflict among Talkers
	111	(Both 101 and 110 conditions)

Because the frame in progress was damaged by the misplaced start_bit (and typically also by the Jam Sequence itself), a frame with something other than a zero_bit in the Jam Detect bit is ignored by all serial bus modules. In this case, no Flag is set or cleared, no Talker considers itself to have sent data, nor does any Listener consider itself to have received data.

La séquence de resynchronisation sert aussi à resynchroniser tous les groupes gestionnaires sur le bus. Typiquement, la trame qui a été invalidée sera retransmise après la resynchronisation du bus série.

Le GESTIONNAIRE DE TRAME dans chaque groupe gestionnaire suit chaque trame en échantillonnant le code de type de trame et en utilisant la valeur échantillonnée pour déterminer le nombre de bits restant dans la trame. Ce nombre de bits atteint, il signale à son EMETTEUR D'EN-TETE associé qu'il n'y a plus de trame en cours. Puisque même une trame de transfert de données de 32 octets n'a que 260 bits après le champ de type de trame, tous les EMETTEURS D'EN-TETE seront prévenus qu'aucune trame n'est en cours, avant la fin de la séquence de resynchronisation de 512 bits.

Lorsqu'un EMETTEUR D'EN-TETE est averti qu'aucune trame n'est en cours, il peut en déclencher une par l'émission d'un bit_départ, par exemple, en présentant XSTART à l'état vrai à son module d'ACCES AU BUS. Cependant, L'EMETTEUR D'EN-TETE surveille également RONE du module d'ACCES AU BUS avant d'émettre un bit_départ. RONE reste à l'état vrai jusqu'à la fin de la séquence de resynchronisation et l'EMETTEUR D'EN-TETE attend la détection de RONE à l'état faux avant d'envoyer le bit_départ. A cet instant, il envoie le bit_départ et procède à l'émission du reste de l'en-tête.

OBSERVATION 3.5:

La procédure décrite ci-dessus sert à la resynchronisation de tous les groupes gestionnaires actifs sur le bus série.

RECOMMANDATION 3.2:

Pour implanter un circuit du bus série qui n'inclut pas (complètement) la fonction de GESTIONNAIRE DE TRAME, il faut le concevoir pour détecter les bit_départ à l'intérieur d'une trame et invalider le bus série, comme décrit dans le chapitre 5 pour un GESTIONNAIRE DE TRAME.

OBSERVATION 3.6:

Cette méthode de conception donne un niveau plus élevé de sécurité et de protection contre les effets parasites qui peuvent être détectés à un point du bus série et pas à un autre.

3.8 *Resynchronisation lors de l'initialisation*

Un groupe gestionnaire peut aussi émettre une séquence de resynchronisation d'au moins 512 bit_un quand il reçoit un signal d'initialisation sur la carte. De cette façon, le bus série peut être utilisé dans les configurations étendues pour lesquelles tous les modules ne partagent pas une ligne d'initialisation commune.

Ainsi, quand un groupe gestionnaire est initialisé, il peut se connecter avec un bus série déjà opérationnel. Au moment où l'initialisation est libérée, le GESTIONNAIRE DE TRAME n'a aucune trace de la trame du bus série, en cours ou non. Par l'émission d'une séquence de resynchronisation, il assure la synchronisation de trame avec les autres gestionnaires.

Ce type de séquence de resynchronisation peut aussi servir comme indicateur aux couches supérieures de gestion du bus série, pour reconfigurer le système du bus série.

The Jam Sequence also serves to resynchronize all the Controller groups on the bus. Typically, the frame which was jammed will be re-sent after the serial bus is resynchronized.

The FRAME MONITOR in each Controller group tracks frames by sampling the Frame Type Code in each one, and using the sampled value to determine the remaining number of bits in the frame. After counting off that number of bits, it then signals its associated HEADER SENDER that a frame is no longer in progress. Since even a 32-byte data transfer frame has only 260 bits after the Frame Type field, all HEADER SENDERS will be informed that no frame is in progress, before the end of the 512-bit Jam Sequence.

Whenever a HEADER SENDER is signalled that a frame is not in progress, it may initiate one by sending a start_bit, i.e. by presenting XSTART True to its BUS ACCESS module. However, the HEADER SENDER also monitors RONE from the BUS ACCESS module before it sends the start_bit. RONE stays True through the end of the Jam Sequence, and the HEADER SENDER waits to send the start_bit until it sees RONE False. At this point it sends the start_bit and proceeds to send the rest of the Header.

OBSERVATION 3.5:

The above procedure serves to resynchronize all the active Controller groups on the serial bus.

RECOMMENDATION 3.2:

When implementing a serial bus circuit that does not include the (complete) FRAME MONITOR function, design it to detect start_bits within a frame and jam the serial bus, as described in Chapter 5 for a FRAME MONITOR.

OBSERVATION 3.6:

This design practice provides greater security and protection against the effects of signal noise that might be sensed at one point on the serial bus, but not at another.

3.8 Jam on Reset

A Controller group is also allowed to send a Jam Sequence of at least 512 one_bits when it receives an on-board Reset signal. In this way, the serial bus can be used in extended configurations in which all the modules do not share a common Reset signal.

So, when a Controller group is Reset, it may be joining an already operating serial bus. At the point when the Reset is released, the Controller's FRAME MONITOR has no tracking of whether a serial bus frame is in progress. By sending a Jam Sequence, it ensures frame synchronization with other Controllers.

This type of Jam Sequence can also serve as an indication to higher layers of serial bus management, to reconfigure the serial bus system.

CHAPITRE 4: COUCHE PHYSIQUE

La couche physique comprend les modules GENERATEUR D'HORLOGE, EXTENSION et ACCES AU BUS. Les modules GENERATEUR D'HORLOGE et EXTENSION assurent la connexion vers les signaux du support physique étendu appelés horloge étendue (EXTCLK+, EXTCLK-) et donnée étendue (EXTDAT+, EXTDAT-). Les modules EXTENSION et ACCES AU BUS assurent la connexion des lignes de signaux du support de fond de panier, appelés horloge série (SERCLK) et donnée série (SERDAT*).

4.1 *Module GENERATEUR D'HORLOGE*

Le module GENERATEUR D'HORLOGE est représenté par la figure 4-1, page 76. Il commande les lignes EXTCLK+ et EXTCLK- du support physique étendu.

REGLE 4.1:

Indifféremment si le bus série est utilisé dans un fond de panier unique ou s'il assure une extension entre plusieurs châssis ou cartes autonomes, il DOIT contenir un et seulement un GENERATEUR D'HORLOGE (actif).

4.1.1 *Interface au support physique*

REGLE 4.2:

SI une carte est connectée à un support physique étendu,
ET qu'elle contient un GENERATEUR D'HORLOGE (actif),
ALORS elle DOIT être située à une extrémité du support physique étendu.

REGLE 4.3:

SI une carte est connectée à un support physique étendu,
ET qu'elle contient un GENERATEUR D'HORLOGE (actif),
ALORS elle DOIT commander en permanence les lignes EXTCLK+ et EXTCLK-, avec un émetteur différentiel ayant les caractéristiques électriques données dans la section 4.4.

RECOMMANDATION 4.1:

Pour permettre l'utilisation sur le même bus série de plusieurs cartes contenant des circuits GENERATEURS D'HORLOGE, les concevoir avec la possibilité de déconnecter l'émetteur EXTCLK du support physique étendu.

CHAPTER 4: THE PHYSICAL LAYER

The Physical Layer includes the CLOCK SOURCE, BRIDGE and BUS ACCESS modules. CLOCK SOURCE and BRIDGE modules connect to the signals in the Extended Medium, namely Extended Clock (EXTCLK+, EXTCLK-) and Extended Data (EXTDAT+, EXTDAT-). BRIDGES and BUS ACCESS modules connect to the signal lines in the Backplane Medium, namely Serial Clock (SERCLK) and Serial Data (SERDAT*).

4.1 *CLOCK SOURCE module*

The CLOCK SOURCE module is shown in Figure 4-1, page 77. It drives the EXTCLK+ and EXTCLK- lines in the Extended Medium.

RULE 4.1:

Regardless of whether a serial bus is used in a single backplane or extends among several subracks and stand-alone boards, it MUST contain one and only one (active) CLOCK SOURCE.

4.1.1 *Medium interface*

RULE 4.2:

IF a board is connected to an Extended Medium,
AND it contains the (active) CLOCK SOURCE,
THEN it MUST be located at one end of the Extended Medium.

RULE 4.3:

IF a board is connected to an Extended Medium,
AND it contains the (active) CLOCK SOURCE,
THEN it MUST drive the EXTCLK+ and EXTCLK- lines with a full-time differential driver with the electrical characteristics given in Section 4.4.

RECOMMENDATION 4.1:

To allow multiple boards containing CLOCK SOURCE hardware to be used on the same serial bus, design them so that the EXTCLK driver can be disconnected from the Extended Medium.

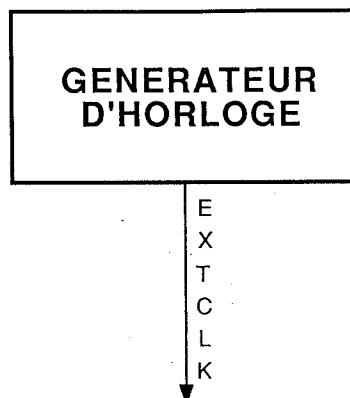
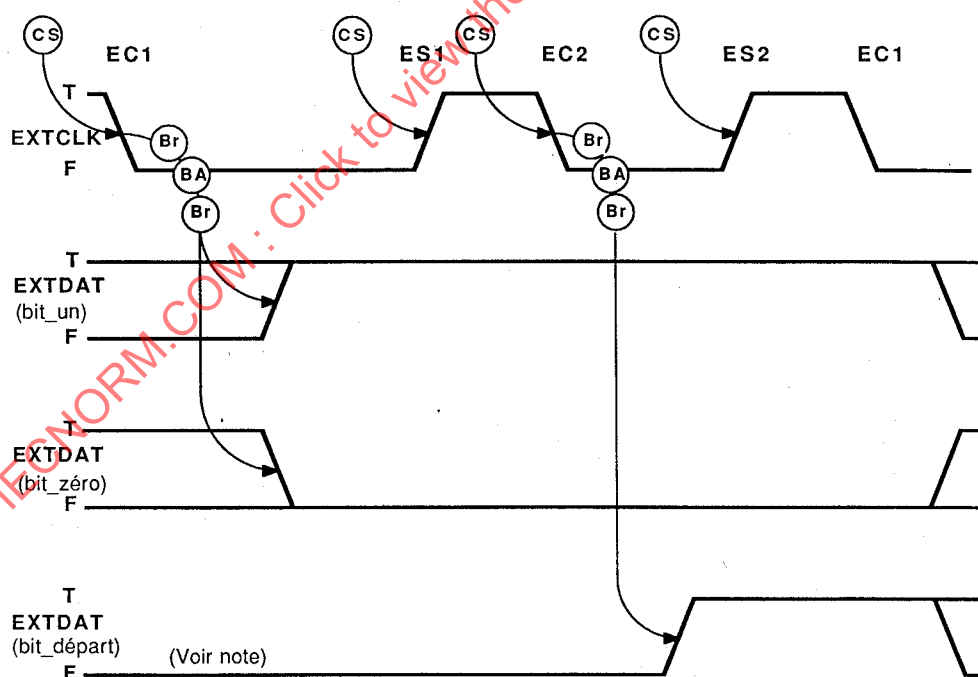


Figure 4-1 - Module GENERATEUR D'HORLOGE



Note:

Comme décrit dans le chapitre 5, un bit_départ est toujours précédé par un bit_zéro.

Figure 4-2 - Formes des signaux du bus d'extension

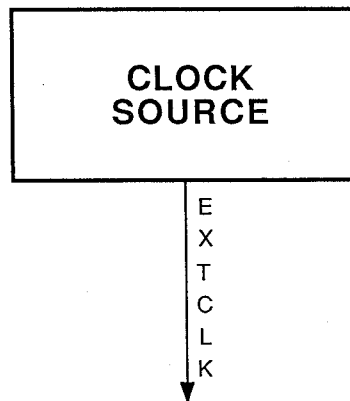
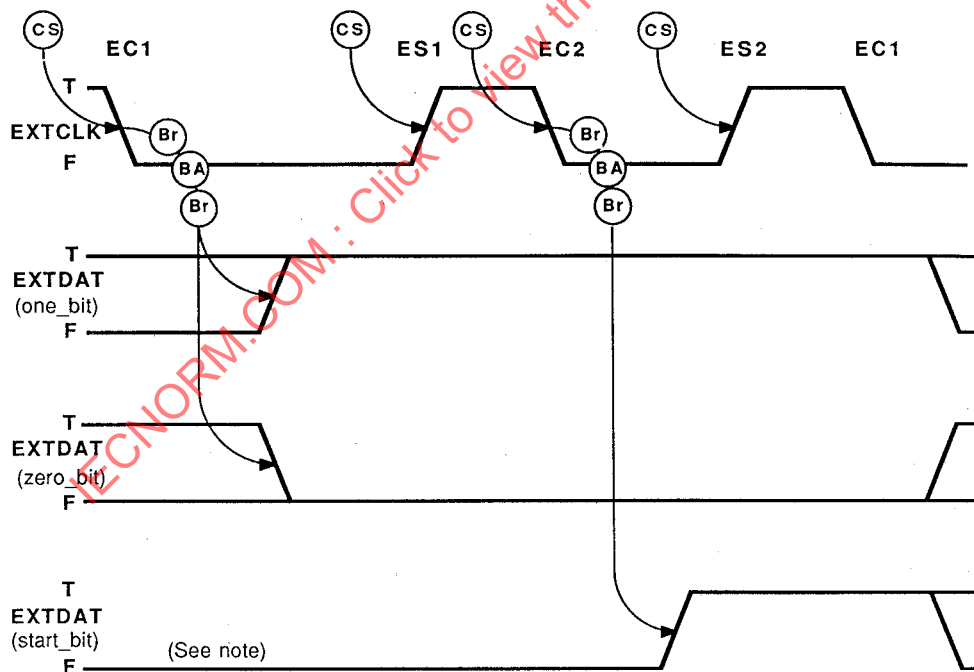


Figure 4-1 - CLOCK SOURCE module

**Note:**

As described in Chapter 5, a start_bit is always preceded by a zero_bit.

Figure 4-2 - Extended bus waveforms

4.1.2 Interface avec la couche physique et la couche de liaison

Le GENERATEUR D'HORLOGE n'a aucune interface avec la couche de liaison de données, ni avec d'autres éléments de la couche physique, en dehors du support physique étendu.

Le paragraphe 4.2.2 analyse des configurations du bus série à châssis unique, dans lequel le "support physique étendu" est constitué seulement par les signaux sur la carte entre un GENERATEUR D'HORLOGE et un module d'EXTENSION.

4.1.3 Fonctionnement

La forme du signal avec laquelle le GENERATEUR D'HORLOGE commande EXTCLK⁺ et EXTCLK⁻ est montrée dans la figure 4-2. La chronologie spécifique de ce signal est fonction de l'utilisation ou non du support physique étendu et, dans ce cas, de sa longueur, spécifiée dans les chapitres 7 et 8.

Les figures 4-2, 4-4 et 4-6, pages 76, 80 et 90, incluent des flèches de cause à effet pour indiquer comment les modules des couches physiques interagissent pour produire les transitions des signaux sur le bus série. Un des symboles:



sur une flèche, indique que les modules GENERATEUR D'HORLOGE, EXTENSION et ACCES AU BUS respectivement, répondent à la transition du départ de la flèche en produisant la transition à la fin de la flèche. Plusieurs symboles sur une flèche signifient que plusieurs modules répondent (de manière séquentielle et dans l'ordre montré) pour produire la transition indiquée.

RECOMMANDATION 4.2:

Pour permettre l'utilisation du support physique étendu de différentes longueurs, concevoir les circuits de GENERATEUR D'HORLOGE pouvant être configurés pour fonctionner avec des vitesses différentes indiquées dans les chapitres 7 et 8.

4.2 Module d'EXTENSION

Le module d'EXTENSION est représenté par la figure 4-3, page 80. Il interconnecte un support physique étendu et un support physique de fond de panier, et convertit les niveaux électriques entre eux. Comme décrit dans cette section, il reconstitue la chronologie sur le fond de panier à partir de celle du support physique étendu.

REGLE 4.4:

Chaque châssis ou carte autonome sur le support physique étendu DOIT avoir un et un seul module d'EXTENSION (actif).

4.2.1 Interfaces avec les supports physiques

Le module d'EXTENSION reçoit les signaux EXTCLK⁺ et EXTCLK⁻ à partir du support physique étendu, et commande le signal SERCLK sur le support physique de fond de panier. Il commande et reçoit les lignes EXTDAT⁺ et EXTDAT⁻ sur le support physique étendu, et commande et reçoit la ligne SERDAT* sur le support physique de fond de panier.

4.1.2 *Physical and Link Layer interfaces*

The CLOCK SOURCE has no interface with the Link Layer, nor with other elements of the Physical Layer, except via the Extended Medium.

Paragraph 4.2.2 discusses single-subrack serial bus configurations, in which the "Extended Medium" consists only of on-board signals between a CLOCK SOURCE and a BRIDGE.

4.1.3 *Operation*

The waveform with which the CLOCK SOURCE drives EXTCLK+ and EXTCLK- is shown in Figure 4-2. The specific timing of this signal is a function of whether or not the Extended Medium is used and, if so, of its length, and is specified in Chapters 7 and 8.

Figures 4-2, 4-4 and 4-6, pages 77, 81 and 91, include cause-and-effect arrows to indicate how the various Physical Layer modules interact to produce signal transitions on the serial bus. One of the symbols:



on an arrow indicates that the CLOCK SOURCE, BRIDGE or BUS ACCESS module, respectively, responds to the transition at the start of the arrow by producing the transition at the end of the arrow. More than one of these symbols on an arrow means that several modules respond (sequentially and in the order shown) to produce the indicated transition.

RECOMMENDATION 4.2:

To allow for use of Extended Media of various lengths, design CLOCK SOURCE hardware so that it can be jumpered to operate at the various speeds cited in Chapters 7 and 8.

4.2 *BRIDGE module*

The BRIDGE module is shown in Figure 4-3, page 81. It interconnects an Extended Medium and a Backplane Medium, and translates the electrical levels between them. As described in this section, it also derives the backplane timing from that on the Extended Medium.

RULE 4.4:

Each subrack or free-standing board on an Extended Medium MUST have one and only one (active) BRIDGE.

4.2.1 *Media interfaces*

The BRIDGE receives the EXTCLK+ and EXTCLK- signals from the Extended Medium, and drives the SERCLK signal in the Backplane Medium. It drives and receives the EXTDAT+ and EXTDAT- lines in the Extended Medium, and drives and receives the SERDAT* line in the Backplane Medium.

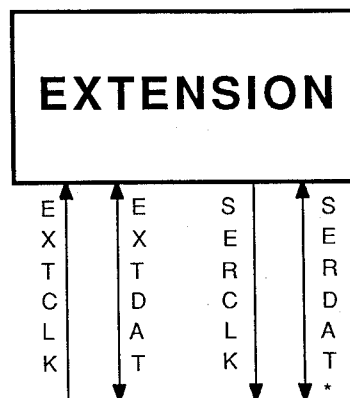


Figure 4-3 - Signaux utilisés par le module d'EXTENSION

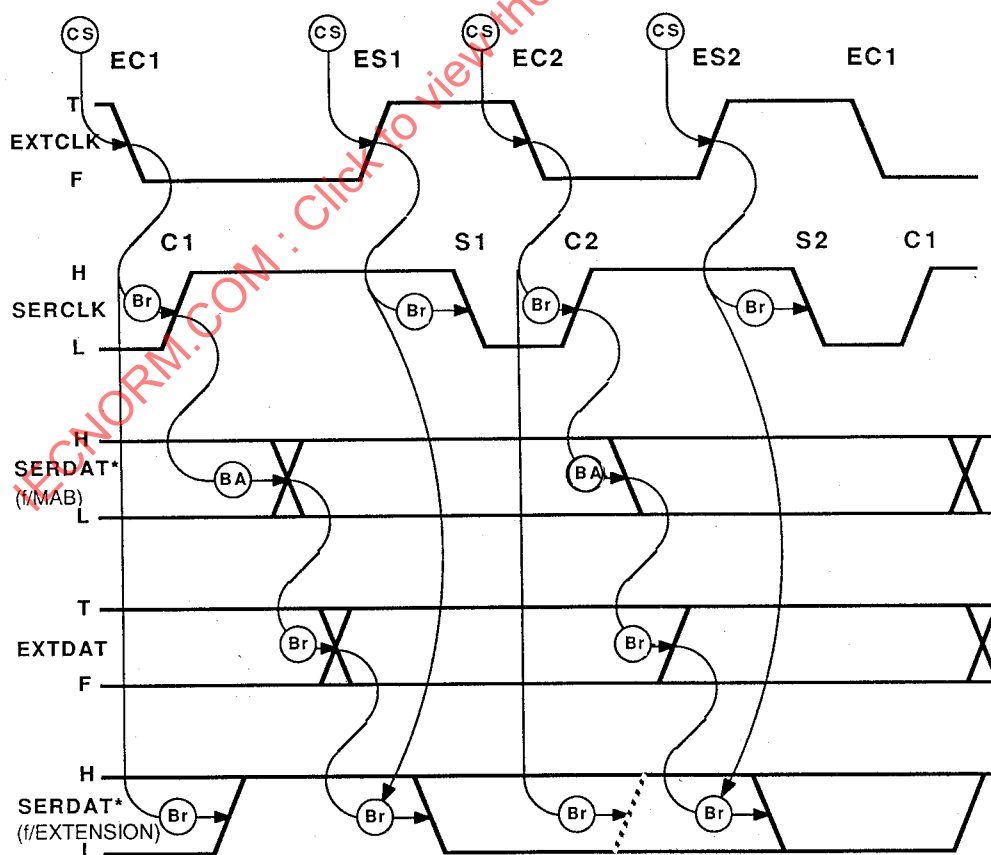


Figure 4-4 - Formes des signaux du bus série

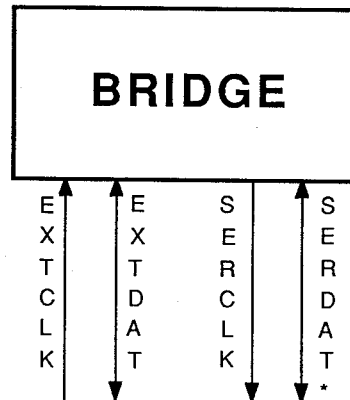


Figure 4-3 - Signals used by the BRIDGE module

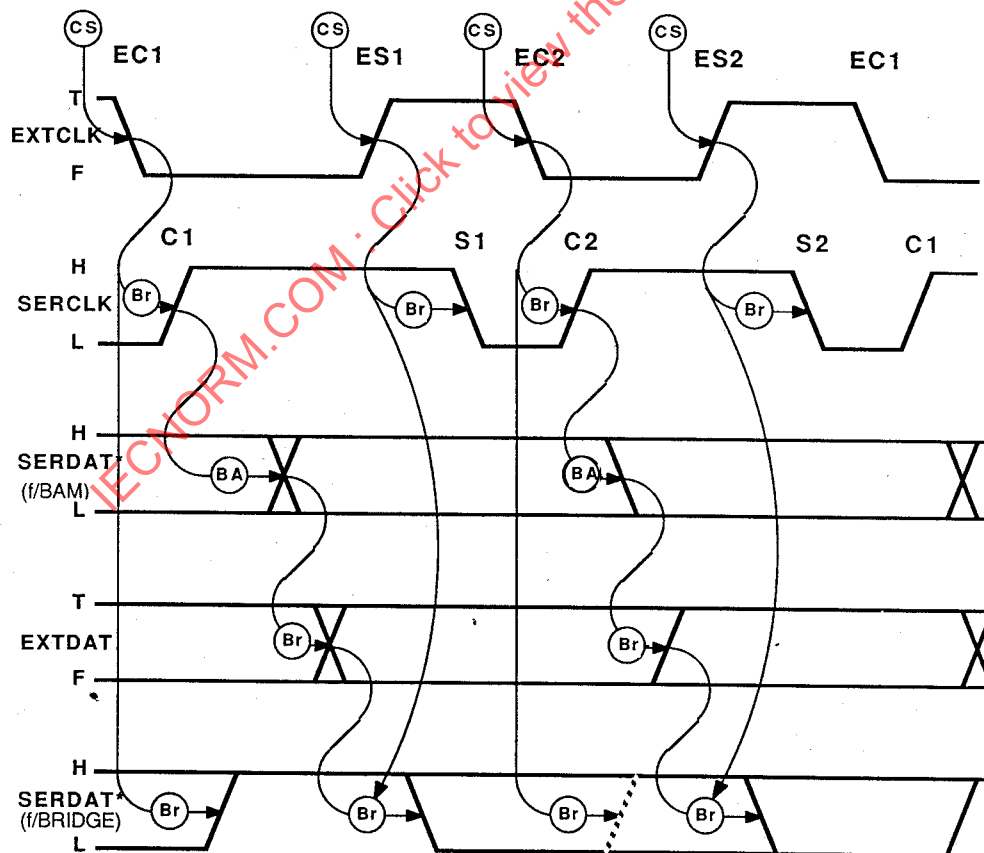


Figure 4-4 - Serial bus waveforms

REGLE 4.5

SI une carte assure une connexion au support physique étendu,
ET elle contient un module d'EXTENSION (actif),
ALORS elle DOIT commander EXTDAT avec un émetteur différentiel fonctionnant à temps partiel avec les caractéristiques électriques données dans la section 4.4.

N'importe quel module d'EXTENSION peut commander EXTDAT à l'état vrai, de telle manière que la tension sur la ligne EXTDAT+ soit plus élevée que la tension sur EXTDAT-. Lorsque aucun module d'EXTENSION ne commande les lignes de cette manière, les résistances d'adaptation du support physique étendu mettent EXTDAT à l'état faux, pour que EXTDAT+ ait une tension inférieure à EXTDAT-.

REGLE 4.6:

Les modules d'EXTENSION NE DOIVENT PAS commander EXTDAT faux, pour que la tension sur EXTDAT+ soit plus basse que celle sur EXTDAT-.

SUGGESTION 4.1:

Afin de satisfaire aux REGLES précédentes, concevoir les modules d'EXTENSION commandant EXTDAT avec un émetteur "RS 485" ayant une entrée de validation. Connecter à l'entrée de validation le signal série de donnée de la carte. Connecter l'entrée normale de donnée de l'émetteur à un niveau logique haut ou bas ou à la donnée série de la carte, comme il est approprié.

REGLE 4.7:

SI une carte est connectée au support physique de fond de panier,
ET elle contient un module d'EXTENSION (actif),
ALORS elle DOIT être située à une extrémité du support physique de fond de panier.

REGLE 4.8

SI une carte est connectée au support physique de fond de panier,
ET elle contient un module d'EXTENSION (actif),
ALORS elle DOIT commander SERCLK avec un émetteur totem-pole compatible TTL à courant élevé ayant les caractéristiques électriques données dans la section 4.4.

REGLE 4.9:

SI une carte est connectée au support physique de fond de panier,
ET elle contient un module d'EXTENSION (actif),
ALORS elle DOIT commander SERDAT* avec un émetteur à collecteur ouvert compatible TTL avec les caractéristiques électriques données dans la section 4.4.

4.2.2 Interfaces avec la couche physique et la couche de liaison

Le module d'EXTENSION n'a aucune interface avec la couche de liaison, ni avec d'autres éléments de la couche physique en dehors du support physique étendu et du support physique de fond de panier.

RULE 4.5:

IF a board provides for Extended Medium connection
AND it contains an (active) BRIDGE,
THEN it MUST drive EXTDAT with a part-time differential driver with the electrical characteristics given in Section 4.4.

Any BRIDGE may drive EXTDAT True, so that the voltage on EXTDAT+ is higher than the voltage on EXTDAT-. When no BRIDGE drives the lines in this way, the Extended Medium's terminating resistors pull EXTDAT False, so that EXTDAT+ has a lower voltage than EXTDAT-.

RULE 4.6:

BRIDGE modules MUST NOT drive EXTDAT False, so that the voltage on EXTDAT+ is lower than that on EXTDAT-.

SUGGESTION 4.1:

To meet the above RULES, design BRIDGE modules to drive EXTDAT with an "RS 485" driver that has an Enable input. Connect the on-board serial data to Enable. Connect the driver's normal data input to a logic high or low level, or to the on-board serial data, as appropriate.

RULE 4.7:

IF a board is connected to a Backplane Medium,
AND it contains the (active) BRIDGE,
THEN the board MUST be located at one end of the Backplane Medium.

RULE 4.8:

IF a board is connected to a Backplane Medium,
AND it contains the (active) BRIDGE,
THEN it MUST drive SERCLK with a TTL-compatible high-current totem pole driver with the electrical characteristics given in Section 4.4.

RULE 4.9:

IF a board is connected to a Backplane Medium,
AND it contains the (active) BRIDGE,
THEN it MUST drive SERDAT* with a TTL-compatible open-collector driver with the electrical characteristics given in Section 4.4.

4.2.2 Physical and Link Layer interfaces

The BRIDGE has no interface with the Link Layer, nor with other elements of the Physical Layer, except via the Extended and Backplane Media.

Le paragraphe 4.3.2 analyse les cartes autonomes du bus série dans lesquelles le support physique de fond de panier est constitué seulement par les signaux sur la carte entre un module d'EXTENSION et un module d'ACCES AU BUS.

AUTORISATION 4.1:

Une carte dont l'utilisation est prévue seulement dans des configurations à châssis unique PEUVENT inclure la fonction de GENERATEUR D'HORLOGE et la fonction de commande de SERCLK du module d'EXTENSION sans fournir une connexion au support physique étendu. Sur une telle carte, le signal logique EXTCLK PEUT être implémenté sans respecter les REGLES concernant le support physique étendu, à condition que les REGLES concernant le support physique de fond de panier soient respectées.

RECOMMANDATION 4.3:

Concevoir les cartes qui commandent SERCLK avec une connexion au support physique étendu. Fournir un GENERATEUR D'HORLOGE séparé qui peut être déconnecté du support physique étendu.

OBSERVATION 4.1:

Concevoir les cartes de cette façon permet leur utilisation dans les deux cas d'application, à châssis unique ou application étendue. En faisant le GENERATEUR D'HORLOGE déconnectable, on permet l'utilisation du produit dans tous les châssis d'un bus étendu.

REGLE 4.10:

SI une carte fournit une connexion au support physique étendu,
ET si elle inclut un GENERATEUR D'HORLOGE et un module d'EXTENSION,
ALORS le module d'EXTENSION DOIT recevoir EXTCLK du support physique étendu et non pas directement du GENERATEUR D'HORLOGE à travers un signal sur la carte.

4.2.3 Fonctionnement.

Chaque "cycle de bit" sur EXTCLK est utilisé pour envoyer un seul bit sur EXTDAT, et inclut quatre transitions appelées EC1, ES1, EC2 et ES2, comme indiqué dans la figure 4-4, page 80. Durant le cycle de bit, le module d'EXTENSION commande EXTDAT à l'état vrai s'il reçoit SERDAT* au niveau bas, et libère EXTDAT s'il reçoit SERDAT* au niveau haut. Les quatre transitions de EXTCLK sont utilisées par le module d'EXTENSION comme suit:

Désignation du front	Utilisation
-------------------------	-------------

EC1	A partir de la transition EXTCLK, un module d'EXTENSION commande SERCLK au niveau haut (générant la transition C1 de SERCLK) et libère son émetteur SERDAT*.
-----	--

ES1	Sur cette transition de EXTCLK, un module d'EXTENSION échantillonne l'état de EXTDAT. S'il échantillonne EXTDAT à l'état vrai, il commande SERDAT* au niveau bas.
-----	---

Paragraph 4.3.2 discusses free-standing serial bus boards, in which the Backplane Medium consists only of on-board signals between a BRIDGE and a BUS ACCESS module.

PERMISSION 4.1:

A board that is intended for use only in single-subrack configurations MAY include the CLOCK SOURCE function and the SERCLK-driving function of the BRIDGE module, without providing an Extended Medium connection. On such a board, the logical signal EXTCLK MAY be implemented without regard for the RULES for the Extended Medium, provided that the RULES for the Backplane Medium are observed.

RECOMMENDATION 4.3:

Design boards that drive SERCLK so that they include a connection to the Extended Medium. Provide a separate CLOCK SOURCE that can be disconnected from the Extended Medium.

OBSERVATION 4.1:

Designing boards in this way allows them to be used in both single-subrack and extended applications. Making the CLOCK SOURCE disconnectable allows the design to be used in all the subracks in an extended bus.

RULE 4.10:

IF a board provides for Extended Medium connection,
AND it includes both a CLOCK SOURCE and a BRIDGE,

THEN the BRIDGE MUST receive EXTCLK from the Extended Medium rather than directly from the CLOCK SOURCE via an on-board signal.

4.2.3 Operation

Each "bit cycle" on EXTCLK is used to send a single bit on EXTDAT and includes four transitions designated EC1, ES1, EC2 and ES2, as shown in Figure 4-4, page 81. Throughout the bit cycle, the BRIDGE module drives EXTDAT True if it receives SERDAT* low, and releases EXTDAT if it receives SERDAT* high. The four EXTCLK transitions are used by the BRIDGE as follows:

Edge Designation	Use
EC1	From this transition of EXTCLK, a BRIDGE drives SERCLK high (making the C1 transition of SERCLK) and releases its SERDAT* driver.
ES1	On this transition of EXTCLK, a BRIDGE samples the state of EXTDAT. If it samples EXTDAT true, it drives SERDAT* low.

Indépendamment de l'état de EXTDAT, le module d'EXTENSION introduit alors un retard d'une durée spécifiée pour permettre à de SERDAT* de se stabiliser, il commande ensuite SERCLK au niveau bas, effectuant la transition S1.

EC2 A partir de cette transition de EXTCLK, un module d'EXTENSION commande SERCLK au niveau haut, effectuant la transition C2 de SERCLK. En plus, il peut libérer l'émetteur de SERDAT*.

ES2 Sur cette transition de EXTCLK, un module d'EXTENSION échantillonne l'état de EXTDAT. S'il échantillonne EXTDAT à l'état vrai, il commande SERDAT* au niveau bas.

Indépendamment de l'état de EXTDAT, le module d'EXTENSION introduit alors un retard d'une durée spécifiée pour permettre à SERDAT* de se stabiliser, il commande ensuite SERCLK au niveau bas, effectuant la transition S2.

4.3 Module d'ACCES AU BUS

Un module d'ACCES AU BUS est montré dans la figure 4-5, page 90. Il interconnecte le support physique de fond de panier avec tous les modules de la couche de liaison de sa carte.

REGLE 4.11:

Une carte NE DOIT PAS inclure plus d'un module d'ACCES AU BUS. Donc, elle NE DOIT PAS contenir plus d'un récepteur sur chacun des signaux SERCLK et SERDAT*, et elle NE DOIT PAS contenir plus d'un émetteur sur SERDAT*.

4.3.1 Interface avec le support physique

Le module d'ACCES AU BUS reçoit les signaux SERCLK et SERDAT* à partir du support physique de fond de panier, et commande aussi SERDAT*.

REGLE 4.12:

SI une carte est connectée à un support physique de fond de panier,
ET elle contient un module d'ACCES AU BUS,
ALORS elle DOIT commander SERDAT* avec un dispositif à collecteur ouvert, compatible TTL, avec les caractéristiques électriques données dans la section 4.4.

4.3.2 Interface avec la couche physique

Les modules d'ACCES AU BUS n'ont aucune interface avec d'autres éléments de la couche physique, excepté par le support physique de fond de panier.

Regardless of the state of EXTDAT, the BRIDGE then delays a specified time to allow SERDAT* to settle, and then drives SERCLK low, making the S1 transition.

EC2 From this transition of EXTCLK, a BRIDGE drives SERCLK high, making the C2 transition of SERCLK. In addition, it may release its SERDAT* driver.

ES2 On this transition of EXTCLK, a BRIDGE samples the state of EXTDAT. If it samples EXTDAT true, it drives SERDAT* low.

Regardless of the state of EXTDAT, the BRIDGE then delays a specified time to allow SERDAT* to settle, and then drives SERCLK low, making the S2 transition.

4.3 BUS ACCESS module

A BUS ACCESS module is shown in Figure 4-5, page 91. It interconnects the Backplane Medium with all of the Link Layer modules on its board.

RULE 4.11:

A board MUST NOT include more than one BUS ACCESS module. That is, it MUST NOT contain more than one receiver each for SERCLK and SERDAT* signals, and it MUST NOT contain more than one driver for SERDAT*.

4.3.1 Medium interface

The BUS ACCESS module receives the SERCLK and SERDAT* signals from the Backplane Medium and also drives SERDAT*.

RULE 4.12:

IF a board is connected to a Backplane Medium,

AND it contains a BUS ACCESS module,

THEN it MUST drive SERDAT* with a TTL-Compatible open-collector device, with the electrical characteristics given in Section 4.4.

4.3.2 Physical Layer interface

BUS ACCESS modules have no interface with other elements of the Physical Layer, except by way of the Backplane Medium.

AUTORISATION 4.2:

Une carte destinée seulement à l'utilisation autonome PEUT être conçue pour inclure les fonctions des modules d'EXTENSION et d'ACCES AU BUS, sans fournir une connexion de support de fond de panier vers les autres cartes. Sur une telle carte, les signaux logiques SERCLK et SERDAT* PEUVENT être implémentés sans tenir compte des REGLES concernant le support physique de fond de panier, à condition que les REGLES concernant le support physique étendu soient respectées.

RECOMMANDATION 4.4:

Même si l'application principale de la carte est l'utilisation autonome, concevoir la carte avec des modules séparés d'EXTENSION et d'ACCES AU BUS et une connexion au support physique de fond de panier vers les autres cartes. Concevoir la carte afin que le module d'EXTENSION puisse être déconnecté du support physique de fond de panier.

OBSERVATION 4.2:

Concevoir les cartes de cette façon permet leur utilisation autonome ou dans un châssis. En réalisant la fonction EXTENSION déconnectable, on permet à plusieurs de ces cartes d'être utilisées dans le même châssis. Cela réduit la charge du support physique étendu lorsque plusieurs cartes "autonomes" sont nécessaires dans une proximité physique réduite.

REGLE 4.13:

SI une carte fournit une connexion au support physique de fond de panier,
ET elle contient les deux modules d'EXTENSION et d'ACCES AU BUS,
ALORS le module d'ACCES AU BUS DOIT recevoir SERCLK du support physique de fond de panier et non pas directement du module d'EXTENSION à travers un signal sur la carte.

PERMISSION 4.2:

A board that is intended only for free-standing use MAY be designed to include the functions of the BRIDGE and BUS ACCESS modules, without providing a Backplane Medium connection to other boards. On such a board, the logical signals SERCLK and SERDAT* MAY be implemented without regard for the RULES for the Backplane Medium, provided that the RULES for the Extended Medium are observed.

RECOMMENDATION 4.4:

Even if the main application of a board is free-standing use, design it with separate BRIDGE and BUS ACCESS modules and a Backplane Medium connection to other boards. Design it so that the BRIDGE function can be disconnected from the Backplane Medium.

OBSERVATION 4.2:

Designing boards in this way allows them to be used alone or in a subrack. Making the BRIDGE function disconnectable allows several such boards to be used in the same subrack. This reduces loading on the Extended Medium when several "free-standing" boards are needed in close physical proximity.

RULE 4.13:

IF a board provides for Backplane Medium connection,

AND it contains both a BRIDGE and a BUS ACCESS module,

THEN the BUS ACCESS module MUST receive SERCLK from the Backplane Medium, rather than directly from the BRIDGE via an on-board signal.

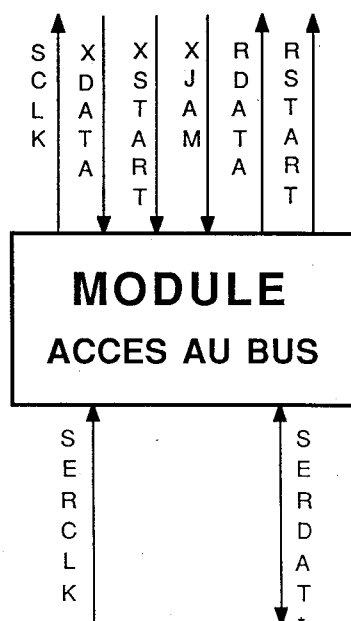
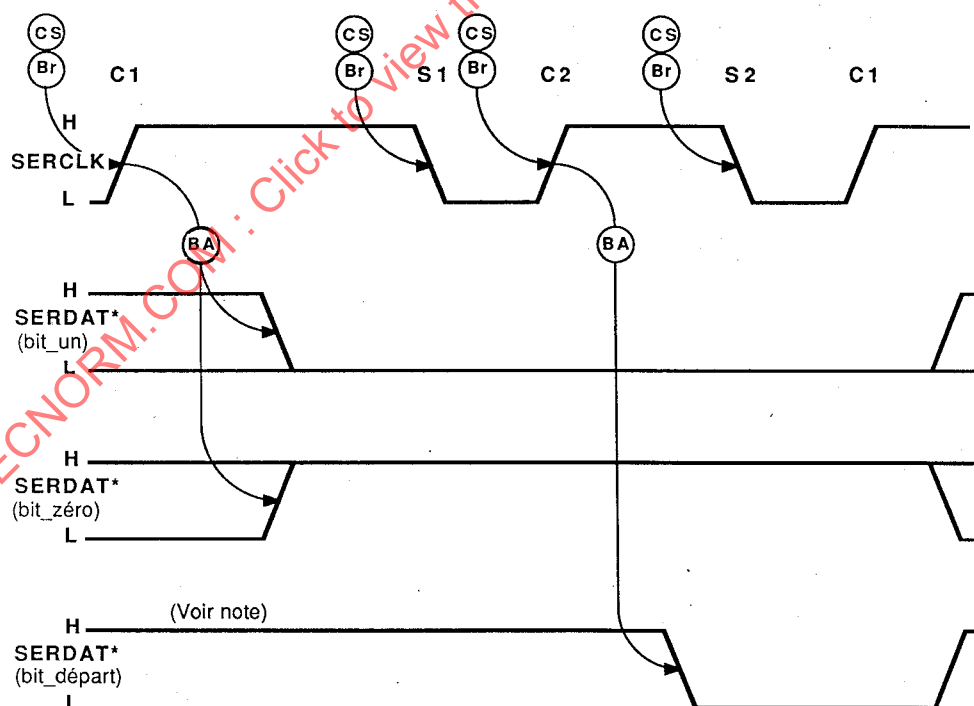


Figure 4-5 - Signaux utilisés par le module d'ACCES AU BUS

**Note:**

Comme décrit dans le chapitre 5, un bit_départ est toujours précédé par un bit_zéro.

Figure 4-6 - Formes des signaux du support physique de fond de panier

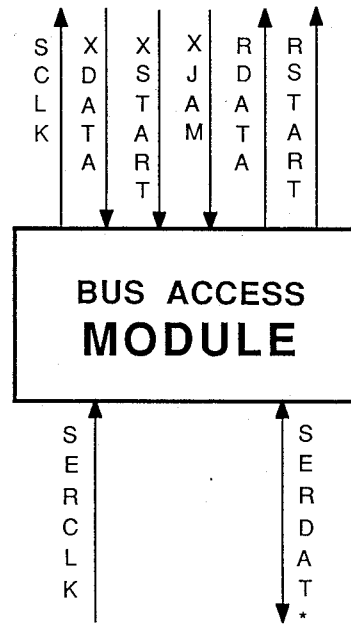
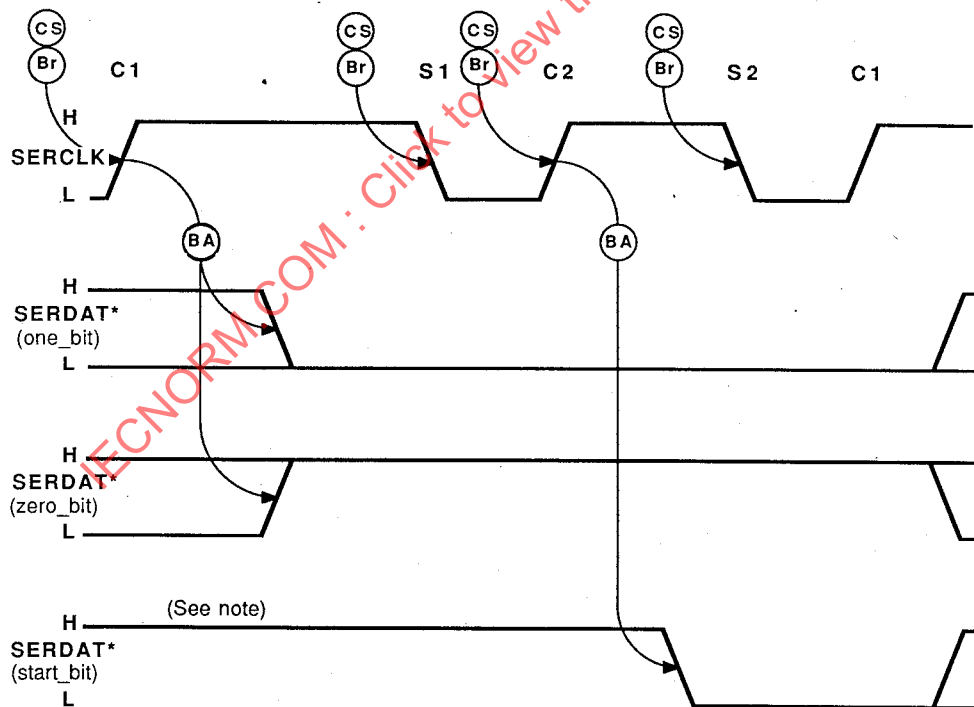


Figure 4-5 - Signals used by the BUS ACCESS module

**Note:**

As described in Chapter 5, a start_bit is always preceded by a zero_bit.

Figure 4-6 - Backplane Medium waveforms

4.3.3 Interface de la couche de liaison

L'interface de la couche de liaison du module d'ACCES AU BUS est formée par les signaux d'entrée XONE, XSTART et XJAM et les signaux de sortie SCLK, RONE et RSTART.

Comme décrit au paragraphe 2.2.1, les modules de la couche de liaison présentent XONE et XSTART à l'état vrai pour indiquer que les modules d'ACCES AU BUS devront commander SERDAT* avec un bit_un ou un bit_départ respectivement. Le module d'ACCES AU BUS commande RONE ou RSTART à l'état vrai pour indiquer respectivement une réception d'un bit_un ou d'un bit_départ, aux modules de la couche de liaison. Les GESTIONNAIRES DE TRAME présentent XJAM à l'état vrai pour indiquer que les modules d'ACCES AU BUS DOIVENT commander SERDAT avec un bit_un de manière combinatoire (immédiat), comme opposé à XONE qui est échantillonné avec le module d'ACCES AU BUS.

SCLK est dérivé de SERCLK par le module d'ACCES AU BUS, et fournit une référence de chronologie pour les modules de la couche de liaison. En réalité, SERCLK est transmis simplement à travers la couche physique pour créer SCLK. Ainsi, la description suivante de la forme de signal de SERCLK s'applique également à SCLK.

OBSERVATION 4.3:

Dans les implémentations du bus série, le module d'ACCES AU BUS a une réalité physique. Il peut être intégré aux fonctions de la couche de liaison, ou il peut être un circuit intégré séparé. Dans les deux cas, il inclut un récepteur SERCLK avec une certaine gamme de temps de propagation. Les concepteurs du matériel de la couche de liaison doivent prendre en compte ce temps de propagation pour atteindre les exigences de performances indiquées dans les chapitres 7 et 8.

Les différents modules de la couche de liaison du bus série *modifient* XONE et XSTART, et *échantillonnent* RONE et RSTART, en fonction des transitions de SCLK.

4.3.4 Fonctionnement

N'importe quel module d'ACCES AU BUS, ou d'EXTENSION, peut commander SERDAT* au niveau bas comme indiqué dans les figures 4-4 et 4-6, pages 80 et 90. Quand aucun module, ni d'ACCES AU BUS ni d'EXTENSION, ne commande SERDAT* au niveau bas, les résistances d'adaptation du support physique de fond de panier positionnent le signal au niveau haut. Le chronogramme spécifique de SERDAT* dépend de l'utilisation ou non du support physique étendu et, dans l'affirmative, est fonction de sa longueur, comme indiqué dans les chapitres 7 et 8.

Chaque "cycle de bit" de SERCLK et SCLK est utilisé pour envoyer un bit unique sur SERDAT*, et inclut quatre transitions désignées par C1, S1, C2 et S2, comme indiqué dans la figure 4-6. Les transitions C1 et C2 sont utilisées par le module d'ACCES AU BUS pour *modifier* les données sur SERDAT*, tandis que les transitions S1 et S2 sont utilisées pour l'*échantillonner*.

4.3.3 Link Layer interface

The Link Layer interface of the BUS ACCESS module consists of the input signals XONE, XSTART and XJAM, and the outputs SCLK, RONE and RSTART.

As described in Paragraph 2.2.1, Link Layer modules present XONE or XSTART True to indicate that the BUS ACCESS modules should drive SERDAT* with a one_bit or a start_bit, respectively. The BUS ACCESS module drives RONE or RSTART True to indicate reception of a one_bit or a start_bit, respectively, to Link Layer modules. FRAME MONITORS present XJAM True to indicate that BUS ACCESS modules should drive SERDAT* with a one_bit in a combinatorial (immediate) fashion, as opposed to XONE which is sampled by the BUS ACCESS module.

SCLK is derived from SERCLK by the BUS ACCESS module, and provides a timing reference to Link Layer modules. Actually, SERCLK is simply passed through the Physical Layer to make SCLK. Thus, the following description of the SERCLK waveform applies equally to SCLK.

OBSERVATION 4.3:

In serial bus implementations, the BUS ACCESS module has a physical reality. It may be integrated with Link Layer functions or it may be a separate IC. In either case, it includes a SERCLK receiver with a possible range of propagation time. Designers of Link Layer hardware need to take this propagation time into account in meeting the performance requirements given in Chapters 7 and 8.

The various Link Layer modules on the serial bus *change* XONE and XSTART, and *sample* RONE and RSTART, based on SCLK transitions.

4.3.4 Operation

Any BUS ACCESS module, or the BRIDGE, may drive SERDAT* Low as shown in Figures 4-4 and 4-6, pages 81 and 91. When no BUS ACCESS module drives SERDAT* Low, and neither does the BRIDGE, the Backplane Medium's terminating resistors pull the signal High. The specific timing of SERDAT* is a function of whether or not the Extended Medium is used and, if so, of its length, and is given in Chapters 7 and 8.

Each "bit cycle" of SERCLK and SCLK is used to send a single bit on SERDAT*, and includes four transitions designated C1, S1, C2 and S2, as shown in Figure 4-6. The C1 and C2 transitions are used by the BUS ACCESS module to *change* the data on SERDAT*, while the S1 and S2 transitions are used to *sample* it.

Désignation du front	Utilisation
C1	Les modules d'ACCES AU BUS échantillonnent XONE sur cette transition et <i>modifient</i> SERDAT* en conséquence.
S1	Les modules d'ACCES AU BUS échantillonnent SERDAT* sur cette transition et modifient en conséquence leurs sorties RONE vers les modules de la couche de liaison.
C2	SI un module d'ACCES AU BUS ne commande pas SERDAT* au niveau bas par un bit_un, ALORS il échantillonne XSTART sur cette transition et <i>modifie</i> SERDAT* en conséquence, SINON il maintient SERDAT* au niveau bas.
S2	SI un module d'ACCES AU BUS n'a pas échantillonné un bit_un sur le front précédent S1, ALORS il échantillonne de nouveau SERDAT* sur cette transition et modifie sa sortie RSTART en conséquence, SINON il maintient (ou positionne) RSTART à l'état faux.

REGLE 4.14:

Les modules d'ACCES AU BUS DOIVENT échantillonner le signal XONE sur la transition C1 de SERCLK et le signal XSTART sur la transition C2, et DOIVENT commander SERDAT* en conséquence, comme indiqué dans le tableau 4-1.

Tableau 4-1 - Protocole de sortie de SERDAT*
pour le module d'ACCES AU BUS

Etat des signaux échantillonnés de la couche de liaison		Type de bit	Commande SERDAT* au niveau bas	
XONE (sur C1)	XSTART (sur C2)		Après C1?	Après C2?
Vrai	x	bit_un	Oui	Oui
Faux	Vrai	bit_départ	Non	Oui
Faux	Faux	bit_zéro	Non	Non

REGLE 4.15:

Les modules d'ACCES AU BUS DOIVENT échantillonner SERDAT* sur les transitions S1 et S2 de SERCLK, et DOIVENT présenter RONE et RSTART en conséquence, comme indiqué dans le tableau 4-2.

Edge designation	Use
C1	BUS ACCESS modules sample XONE on this transition and <i>change</i> SERDAT* accordingly.
S1	BUS ACCESS modules <i>sample</i> SERDAT* on this transition and change their RONE outputs to Link Layer modules accordingly.
C2	IF a BUS ACCESS module is not driving SERDAT* Low for a one_bit, THEN it samples XSTART on this transition and <i>changes</i> SERDAT* accordingly, ELSE it maintains SERDAT* Low.
S2	IF a BUS ACCESS module did not sample a one_bit at the preceding S1 edge, THEN it <i>samples</i> SERDAT* again on this transition and changes its RSTART output accordingly, ELSE it keeps (or makes) RSTART False.

RULE 4.14:

BUS ACCESS modules MUST sample the XONE signal at the C1 transition of SERCLK and the XSTART signal at the C2 transition, and MUST drive SERDAT* accordingly, as shown in Table 4-1.

Table 4-1 - SERDAT* output protocol for BUS ACCESS module

State of sampled Link layer signals		Type of bit	Drive SERDAT* Low	
XONE (at C1)	XSTART (at C2)		After C1?	After C2?
True	x	One_bit	Yes	Yes
False	True	Start_bit	No	Yes
False	False	Zero_bit	No	No

RULE 4.15:

BUS ACCESS modules MUST sample SERDAT* at the S1 and S2 transitions of SERCLK, and MUST present RONE and RSTART accordingly, as shown in Table 4-2.

**Tableau 4-2 - Protocole d'entrée de SERDAT*
pour le module d'ACCES AU BUS**

Niveau de SERDAT* échantillonné		Type de bit	Etat du signal de la couche de liaison	
sur S1	sur S2		RONE (après S1)	RSTART (après S2)
Bas	x	bit_un	Vrai	Faux
Haut	Bas	bit_départ	Faux	Vrai
Haut	Haut	bit_zéro	Faux	Faux

OBSERVATION 4.4:

Un "x" dans les tableaux 4-1 et 4-2 indique que l'état du signal d'entrée n'affecte pas l'opération du module dans ce cas.

REGLE 4.16:

SI un GESTIONNAIRE DE TRAME présente XJAM à l'état vrai,
ALORS son module d'ACCES AU BUS DOIT commander SERDAT* au
niveau bas sans attendre une transition de SERCLK/SCLK.

4.4 Spécifications électriques

4.4.1 Caractéristiques d'entrée du support physique de fond de panier

REGLE 4.17:

Les récepteurs pour SERCLK et SERDAT* DOIVENT fournir des limitations afin de prévenir des excursions négatives en dessous de -1,5 V.

OBSERVATION 4.5:

Les circuits standard 74LSxxx et 74Fxxx ont des diodes de limitation interne sur les entrées qui satisferont à la REGLE ci-dessus.

REGLE 4.18:

Les récepteurs pour SERCLK et SERDAT* DOIVENT détecter un niveau logique haut quand la ligne est au-dessus de 2,0 V.

REGLE 4.19:

Les récepteurs pour SERCLK et SERDAT* DOIVENT détecter un niveau logique bas quand le niveau sur la ligne est en dessous de 0,8 V.

RECOMMANDATION 4.5:

Utiliser des récepteurs qui incluent un hystérésis entre les zones de seuil côté négatif et côté positif.

Table 4-2 - SERDAT* input protocol for BUS ACCESS module

SERDAT* level sampled		Type of bit	Link layer signal state	
at S1	at S2		RONE (after S1)	RSTART (after S2)
Low	x	One_bit	True	False
High	Low	Start_bit	False	True
High	High	Zero_bit	False	False

OBSERVATION 4.4:

An "x" in Tables 4-1 and 4-2 indicates that the state of the input signal does not affect the operation of the module in that case.

RULE 4.16:

IF a FRAME MONITOR presents XJAM True,
THEN its BUS ACCESS module MUST drive SERDAT* low without waiting for a SERCLK/SCLK transition.

4.4 Electrical specifications**4.4.1 Input characteristics for the Backplane Medium****RULE 4.17:**

Receivers for SERCLK and SERDAT* MUST provide clamping that prevents negative excursions below -1.5 V.

OBSERVATION 4.5:

Standard 74LSxxx and 74Fxxx devices have internal clamping diodes on their inputs that will satisfy the above RULE.

RULE 4.18:

Receivers for SERCLK and SERDAT* MUST detect a high logic level when the line is above 2.0 V

RULE 4.19:

Receivers for SERCLK and SERDAT* MUST detect a low logic level when the line is below 0.8 V.

RECOMMENDATION 4.5:

Use receivers that include hysteresis between their negative-going and positive-going threshold points.

4.4.2 Commande et charge pour SERCLK

REGLE 4.20:

Les émetteurs SERCLK DOIVENT satisfaire aux spécifications suivantes:

- Courant absorbé au niveau bas: $IOL \geq 64 \text{ mA}$.
- Tension au niveau bas: $VOL \leq 0,6 \text{ V}$ à $IOL = 64 \text{ mA}$.
- Courant fourni au niveau haut: $IOH \geq 3 \text{ mA}$.
- Tension au niveau haut: $VOH \geq 2,4 \text{ V}$ à $IOH = 3 \text{ mA}$.
- Courant source minimal avec la broche SERCLK à la masse: $IOS \geq 50 \text{ mA}$ à 0 V .
- Courant source maximal avec la broche SERCLK à la masse: $IOS \leq 225 \text{ mA}$ à 0 V .

REGLE 4.21:

Une carte DOIT limiter sa charge sur SERCLK aux valeurs suivantes:

- Courant fourni par la carte à $0,6 \text{ V}$: $IL \leq 400 \mu\text{A}$.
- Courant absorbé par la carte à $2,4 \text{ V}$: $IH \leq 50 \mu\text{A}$.
- Charge capacitive totale, incluant la piste du signal:
 - Carte avec module d'EXTENSION: $CT \leq 20 \text{ pF}$.
 - Autres cartes: $CT \leq 12 \text{ pF}$.

4.4.3 Commande et charge pour SERDAT*

REGLE 4.22:

Un émetteur SERDAT* DOIT satisfaire aux spécifications suivantes:

- Courant absorbé au niveau bas: $IOL \geq 60 \text{ mA}$.
- Tension de niveau bas: $VOL \leq 0,6 \text{ V}$ à $IOL = 60 \text{ mA}$.

REGLE 4.23:

Une carte DOIT limiter sa charge sur SERDAT* aux valeurs suivantes:

- Courant émis par la carte à $0,6 \text{ V}$: $IIL \leq 400 \mu\text{A}$.
- Courant absorbé par la carte à $2,4 \text{ V}$: $IIH \leq 70 \mu\text{A}$.
- Charge capacitive totale, incluant la piste du signal: $CT \leq 20 \text{ pF}$.

OBSERVATION 4.6:

Les courants absorbés ci-dessus incluent à la fois le courant du récepteur et le courant de fuite de l'émetteur.

4.4.2 Driving and loading for SERCLK

RULE 4.20:

SERCLK drivers MUST meet the following specifications:

- Low state sink current: $IOL \geq 64 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 64 \text{ mA}$.
- High state source current: $IOH \geq 3 \text{ mA}$.
- High state voltage: $VOH \geq 2.4 \text{ V}$ at $IOH = 3 \text{ mA}$.
- Minimum source current with SERCLK pin grounded: $IOS \geq 50 \text{ mA}$ at 0 V .
- Maximum source current with SERCLK pin grounded: $IOS \leq 225 \text{ mA}$ at 0 V .

RULE 4.21:

A board MUST limit its loading on SERCLK to the following values:

- Current sourced by board 0.6 V : $IL \leq 400 \text{ }\mu\text{A}$.
- Current sunk by board at 2.4 V : $IH \leq 50 \text{ }\mu\text{A}$.
- Total capacitive load, including signal trace:
- Board with BRIDGE module: $CT \leq 20 \text{ pF}$.
- Other boards: $CT \leq 12 \text{ pF}$.

4.4.3 Driving and loading for SERDAT*

RULE 4.22:

A SERDAT* driver MUST meet the following specifications:

- Low state sink current: $IOL \geq 60 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 60 \text{ mA}$.

RULE 4.23:

A board MUST limit its loading on SERDAT* to the following values:

- Current sourced by board at 0.6 V : $IIL \leq 400 \text{ }\mu\text{A}$.
- Current sunk by board at 2.4 V : $IIH \leq 70 \text{ }\mu\text{A}$.
- Total capacitive load, including signal trace: $CT \leq 20 \text{ pF}$.

OBSERVATION 4.6:

The sink current listed above includes both receiver current and driver leakage current.

4.4.4 Caractéristiques d'entrée pour le support physique étendu

REGLE 4.24:

Les récepteurs de EXTCLK et EXTDAT* DOIVENT détecter l'état vrai quand la tension sur la ligne "+" est supérieure de 0,2 V à celle de la ligne "-", étant donné que les deux lignes sont entre -7 V et +12 V, par rapport à la masse du dispositif.

REGLE 4.25:

Les récepteurs de EXTCLK et EXTDAT* DOIVENT détecter l'état faux quand la tension sur la ligne "-" est supérieure de 0,2 V à la ligne "+", étant donné que les deux lignes sont entre -7 V et +12 V, par rapport à la masse du dispositif.

4.4.5 Commande du support physique étendu

REGLE 4.26:

Les émetteurs de EXTCLK et EXTDAT DOIVENT garantir les spécifications suivantes quand ils sont connectés au circuit de charge de test de la figure 4-7, page ci-dessous:

- Tension différentielle de l'état vrai: $1,5 \text{ V} \leq V_{abt} \leq 3,5 \text{ V}$.
- Tension de décalage de l'état vrai: $2,0 \text{ V} \leq V_{ost} \leq 3,0 \text{ V}$.

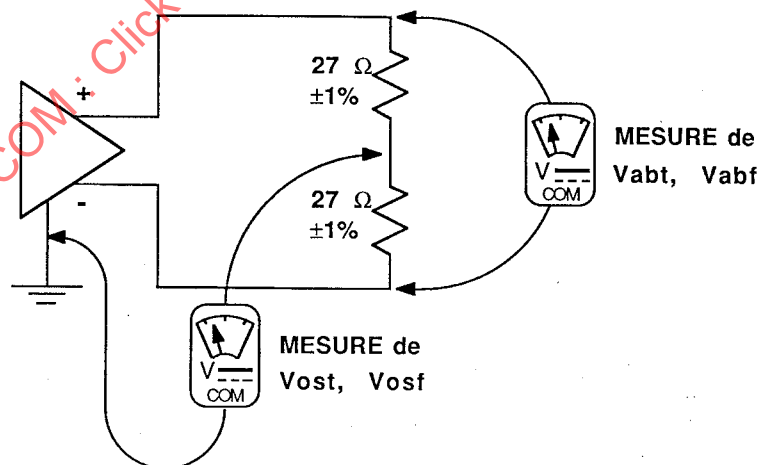


Figure 4-7 - Circuit de charge de test pour la REGLE 4.26 et la REGLE 4.27

4.4.4 Input characteristics for the Extended Medium

RULE 4.24:

Receivers for EXTCLK and EXTDAT* MUST detect the True state when the voltage on the "+" line is more than 0.2 V above that on the "-" line, provided that both lines are between -7 V and +12 V, referenced to the device ground pin.

RULE 4.25:

Receivers for EXTCLK and EXTDAT* MUST detect the False state when the voltage on the "-" line is more than 0.2 V above that on the "+" line, provided that both lines are between -7 V and +12 V, referenced to the device ground pin.

4.4.5 Driving the Extended Medium

RULE 4.26:

Drivers for EXTCLK and EXTDAT MUST guarantee the following specifications when connected to the test load circuit of Figure 4-7 below:

- True state differential voltage: $1.5 \text{ V} \leq V_{abt} \leq 3.5 \text{ V}$.
- True state offset voltage: $2.0 \text{ V} \leq V_{ost} \leq 3.0 \text{ V}$.

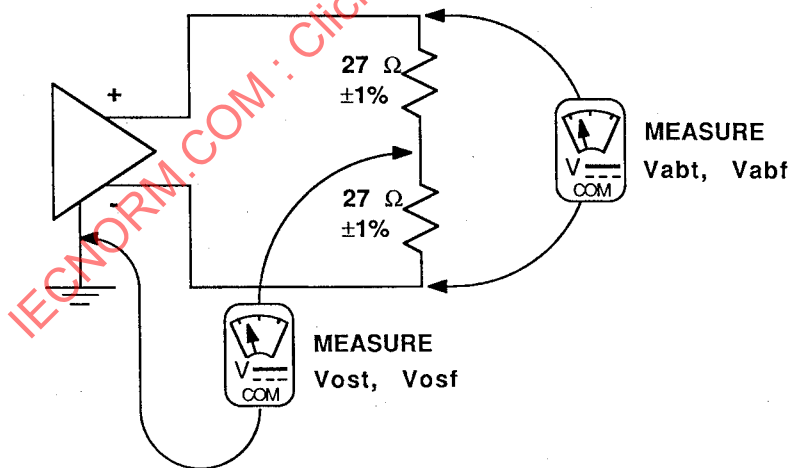


Figure 4-7 - Test load circuit for RULE 4.26 and RULE 4.27

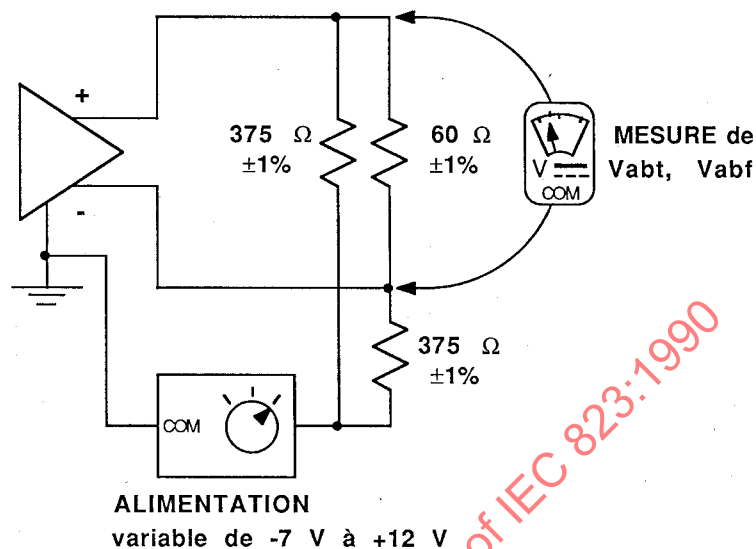


Figure 4-8 - Circuit de charge de test pour la REGLE 4.28 et la REGLE 4.29

REGLE 4.27:

Les émetteurs de EXTCLK DOIVENT garantir les spécifications additionnelles suivantes quand ils sont connectés au circuit de charge de test de la figure 4-7, page 100:

- Tension différentielle d'état faux: $-3,5 \text{ V} \leq V_{abf} \leq -1,5 \text{ V}$.
- Tension de décalage d'état faux: $2,0 \text{ V} \leq V_{osf} \leq 3,0 \text{ V}$.
- F/V différentielle à différentielle: $|V_{abt} + V_{abf}| \leq 0,2 \text{ V}$.
(Noter que V_{abf} est négative)
- F/V décalage à décalage: $|V_{ost} - V_{osf}| \leq 0,2 \text{ V}$.

REGLE 4.28:

Les émetteurs de EXTCLK et EXTDAT DOIVENT garantir la spécification suivante quand ils sont connectés au circuit de test de la figure 4-8, ci-dessus, pour toutes les valeurs de la source de tension V_{tst} entre -7 V et $+12 \text{ V}$:

- Tension différentielle d'état vrai: $1,5 \text{ V} \leq V_{abt} \leq 3,5 \text{ V}$.

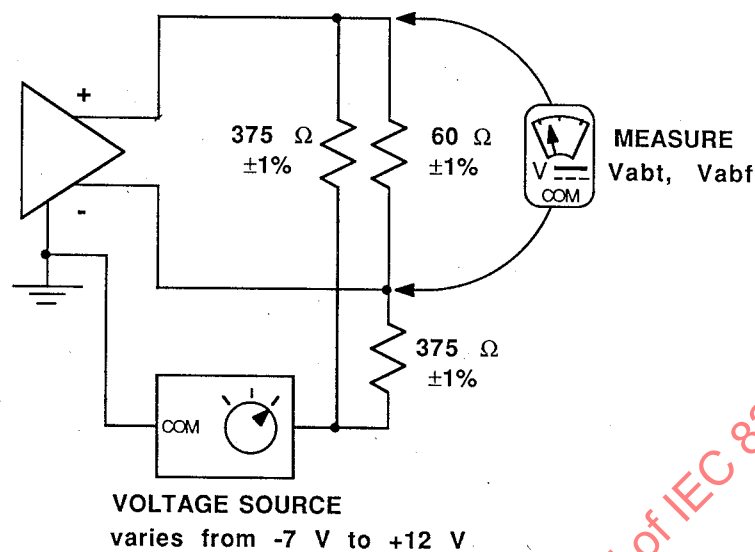


Figure 4-8 - Test load circuit for RULE 4.28 and RULE 4.29

RULE 4.27:

Drivers for EXTCLK MUST guarantee the following additional specifications when connected to the test load circuit of Figure 4-7, page 101:

- False state differential voltage: $-3.5 \text{ V} \leq V_{abf} \leq -1.5 \text{ V}$.
- False state offset voltage: $2.0 \text{ V} \leq V_{osf} \leq 3.0 \text{ V}$.
- F/T differential differential: $|V_{abt} + V_{abf}| \leq 0.2 \text{ V}$.
(Note that V_{abf} is negative)
- F/T offset offset: $|V_{ost} - V_{osf}| \leq 0.2 \text{ V}$.

RULE 4.28:

Drivers for EXTCLK and EXTDAT MUST guarantee the following specification when connected to the test circuit of Figure 4-8, above, for all values of the voltage source V_{tst} between -7 V and +12 V:

- True state differential voltage: $1.5 \text{ V} \leq V_{abt} \leq 3.5 \text{ V}$.

REGLE 4.29:

Les émetteurs de EXTCLK DOIVENT garantir les spécifications additionnelles suivantes quand ils sont connectés au circuit test de la figure 4-8, pour toutes les valeurs de source de tension V_{tst} entre -7 V et +12 V:

- Tension différentielle d'état faux: $-3,5 \text{ V} \leq V_{abf} \leq -1,5 \text{ V}$.
- F/V différentielle à différentielle: $|V_{abt} + V_{abf}| \leq 0,2 \text{ V}$.
(noter que V_{abf} est négative)

4.4.6 Charge du support physique étendu

REGLE 4.30:

Les cartes DOIVENT avoir une balance de charge courant continu sur les deux lignes de chaque paire EXTCLK et EXTDAT, comme suit:

- SI les deux entrées d'une paire sont connectées ensemble à un circuit de test,
 ET que ces entrées en court-circuit sont connectées à une source de tension entre -7 V et +12 V,
 ALORS le courant émis ou absorbé entre ces entrées en court-circuit et la source de tension DOIT être plus petit que 0,2 mA.

REGLE 4.31:

Les cartes DOIVENT limiter leur charge courant continu sur EXTCLK et EXTDAT comme suit:

- SI dans un circuit test, une des deux lignes d'une paire est reliée à la masse,
 ET que la tension sur l'autre (V) varie de -7 V à +12 V,
 ALORS le courant émis/absorbé par la carte sur cette ligne (I) DOIT s'insérer dans une gamme applicable suivante:

$$V = +5 \text{ V à } +12 \text{ V} \quad 0 \leq I \leq 67 (V+3) \mu\text{A}$$

$$V = -3 \text{ V à } +5 \text{ V} \quad 67 (V-5) \mu\text{A} \leq I \leq 67 (V+3) \mu\text{A}$$

$$V = -7 \text{ V à } -3 \text{ V} \quad 67 (V-5) \mu\text{A} \leq I \leq 0$$

REGLE 4.32:

Les cartes DOIVENT limiter leurs charges en courant alternatif sur le support physique étendu de telle manière que:

- SI dans un circuit de test, un signal d'une paire est connecté à a masse,
 ALORS la capacité de l'autre signal par rapport à la masse, incluant la piste du signal et l'émetteur et/ou le récepteur, DOIT être inférieure ou égale à 20pF.

RULE 4.29:

Drivers for EXTCLK MUST guarantee the following additional specifications when connected to the test circuit of Figure 4-8, for all values of the voltage source V_{tst} between -7 V and +12 V:

- False state differential voltage: $-3.5 \text{ V} \leq V_{abf} \leq -1.5 \text{ V}$.
- F/T differential differential: $|V_{abt} + V_{abf}| \leq 0.2 \text{ V}$.
(Note that V_{abf} is negative))

4.4.6 Loading on the Extended Medium

RULE 4.30:

Boards MUST have balanced d.c. loading on the two lines of each of the EXTCLK and EXTDAT pairs, as follows:

IF the two inputs of a pair are connected together in a test circuit,
AND these short-circuited inputs are connected to a voltage source between 7 V and +12 V,
THEN the current sourced or sunk between these short-circuited inputs and the voltage source MUST be less than 0.2 mA.

RULE 4.31:

Boards MUST limit their d.c. loading on EXTCLK and EXTDAT as follows:

IF in a test circuit, either of the two lines in a pair is grounded,
AND the voltage on the other (V) is varied between -7 V and +12 V,
THEN the current sourced/sunk by the board on the latter line (I) MUST fall within the applicable range among the following:

$$V = +5 \text{ V to } +12 \text{ V} \quad 0 \leq I \leq 67 (V+3) \text{ } \mu\text{A}$$

$$V = -3 \text{ V to } +5 \text{ V} \quad 67 (V-5) \text{ } \mu\text{A} \leq I \leq 67 (V+3) \text{ } \mu\text{A}$$

$$V = -7 \text{ V to } -3 \text{ V} \quad 67 (V-5) \text{ } \mu\text{A} \leq I \leq 0$$

RULE 4.32:

Boards MUST limit their a.c. loading on the Extended Medium such that:

IF in a test circuit, one signal of a pair is grounded,
THEN the capacitance of the other signal with respect to ground, including the signal trace and the driver and/or receiver, MUST be 20 pF or less.

CHAPITRE 5: MODULES DE LA COUCHE DE LIAISON DE DONNEES

Un système de bus série comprend, pour la couche de liaison de données, les modules fonctionnels suivants:

EMETTEURS D'EN-TETE
RECEPTEUR D'EN-TETE
EMETTEURS DE DONNEE
RECEPTEURS DE DONNEE
GESTIONNAIRE DE TRAME

OBSERVATION 5.1:

Ces "modules fonctionnels" sont utilisés comme support à la discussion du protocole du bus série, et il ne faut pas les considérer comme une contrainte pour la conception de la logique. Par exemple, un seul circuit intégré peut réaliser plusieurs de ces fonctions.

Les sections suivantes décrivent le comportement de ces modules, y compris leur interaction avec les couches supérieures du gestionnaire du bus série.

OBSERVATION 5.2:

Il est probable que les circuits du bus série soient conçus avec des signaux sur les cartes et des architectures internes variés. Cette norme décrit parfois certains aspects de l'interface avec les couches supérieures en termes généraux. Par exemple, on peut dire que "les couches supérieures lisent à partir d'un RECEPTEUR DE DONNEES" sans spécifier exactement comment cela est fait. Cela pour éviter de placer des contraintes inutiles pour la conception de la logique du matériel d'interface avec le bus série. Dans certains cas, les lignes réelles de signal sont montrées pour améliorer l'explication du bus série. Quand cela est fait, le but est de préciser quelle information transite entre le module et une autre logique sur la carte. D'autres méthodes pour transférer cette information sont aussi possibles.

OBSERVATION 5.3:

On peut s'attendre à ce que des circuits conçus pour le bus série comprennent une partie de ce qui est décrit dans cette norme comme de la "logique des couches supérieures". Par exemple, les modules fonctionnels décrits ci-après ont plusieurs sorties qui fournissent une impulsion dont la durée est de 1 cycle de SCLK. Les circuits réels peuvent utiliser ces "impulsions conceptuelles" pour fournir des sorties de durée plus longue.

OBSERVATION 5.4:

Les concepteurs de matériels du bus série doivent s'assurer qu'il n'y aura aucun problème de métastabilité sur SERDAT* résultant de l'échantillonnage des entrées venant des couches supérieures du gestionnaire du bus série.

4.4.7 Stress/damage resistance for the Extended Medium

RULE 4.33:

Drivers and Receivers for EXTCLK and EXTDAT MUST guarantee that they will not be damaged by any combination of voltages on the "+" and "-" lines, each voltage being between -7 V and +12 V (referenced to the device ground pin). Further, they MUST guarantee this for both the power on and power-off states and, for a driver, with the driver turned on or off.

RULE 4.34:

Drivers for EXTCLK and EXTDAT MUST guarantee that they will not be damaged by a short circuit between the "+" and "-" lines, of any duration.

RULE 4.35:

Drivers and Receivers for EXTCLK and EXTDAT MUST guarantee that they will not be damaged by any combination of pulses, each pulse being up to ± 25 V in amplitude (referenced to the device ground pin) and up to 15 μ s in duration, provided that the pulses do not add up to more than 100 ms duration in any 10 s period, and provided that the pulse source has at least a 100 Ω series source impedance between the "+" or "-" line and the device ground pin. Further, the devices MUST guarantee this damage resistance in both the power-on and power-off state and, for a driver, with the driver turned on or off.

IECNORM.COM : Click to view the full PDF of IEC 823:1990

CHAPITRE 5: MODULES DE LA COUCHE DE LIAISON DE DONNEES

Un système de bus série comprend, pour la couche de liaison de données, les modules fonctionnels suivants:

EMETTEURS D'EN-TETE
RECEPTEUR D'EN-TETE
EMETTEURS DE DONNEE
RECEPTEURS DE DONNEE
GESTIONNAIRE DE TRAME

OBSERVATION 5.1:

Ces "modules fonctionnels" sont utilisés comme support à la discussion du protocole du bus série, et il ne faut pas les considérer comme une contrainte pour la conception de la logique. Par exemple, un seul circuit intégré peut réaliser plusieurs de ces fonctions.

Les sections suivantes décrivent le comportement de ces modules, y compris leur interaction avec les couches supérieures du gestionnaire du bus série.

OBSERVATION 5.2:

Il est probable que les circuits du bus série soient conçus avec des signaux sur les cartes et des architectures internes variés. Cette norme décrit parfois certains aspects de l'interface avec les couches supérieures en termes généraux. Par exemple, on peut dire que "les couches supérieures lisent à partir d'un RECEPTEUR DE DONNEES" sans spécifier exactement comment cela est fait. Cela pour éviter de placer des contraintes inutiles pour la conception de la logique du matériel d'interface avec le bus série. Dans certains cas, les lignes réelles de signal sont montrées pour améliorer l'explication du bus série. Quand cela est fait, le but est de préciser quelle information transite entre le module et une autre logique sur la carte. D'autres méthodes pour transférer cette information sont aussi possibles.

OBSERVATION 5.3:

On peut s'attendre à ce que des circuits conçus pour le bus série comprennent une partie de ce qui est décrit dans cette norme comme de la "logique des couches supérieures". Par exemple, les modules fonctionnels décrits ci-après ont plusieurs sorties qui fournissent une impulsion dont la durée est de 1 cycle de SCLK. Les circuits réels peuvent utiliser ces "impulsions conceptuelles" pour fournir des sorties de durée plus longue.

OBSERVATION 5.4:

Les concepteurs de matériels du bus série doivent s'assurer qu'il n'y aura aucun problème de métastabilité sur SERDAT* résultant de l'échantillonnage des entrées venant des couches supérieures du gestionnaire du bus série.

CHAPTER 5: LINK LAYER MODULES

A serial bus system includes the following Link Layer functional modules:

HEADER SENDERS
HEADER RECEIVERS
DATA SENDERS
DATA RECEIVERS
FRAME MONITORS

OBSERVATION 5.1:

These "functional modules" are used as vehicles for discussion of the serial bus protocol, and need not be considered a constraint to logic design. For example, a single integrated circuit (IC) may provide several of these functions.

The following sections describe the behaviour of these modules, including their interaction with higher layers of serial bus management.

OBSERVATION 5.2:

It is likely that circuits for the serial bus will be designed with various internal architectures and on-board signals. This standard sometimes describes some aspects of higher-layer interfaces in general terms. For instance, we may say that "higher layers read from a DATA RECEIVER" without specifying exactly how this is accomplished. This is done to avoid placing unnecessary constraints on the logic design of serial bus interface hardware. In some cases actual signal lines are shown, to improve the explanation of the serial bus. Where this is done, the intent is to make clear what information crosses the boundary between the module and other on-board logic. Other methods for conveying this same information across the boundary are also permissible.

OBSERVATION 5.3:

It is expected that circuits designed for the serial bus will include some of what is described in this standard as "higher-layer logic". For example, the functional modules described below have several outputs which produce a pulse which is 1 SCLK cycle in duration. Real circuits might use these "conceptual pulses" to produce outputs of longer duration.

OBSERVATION 5.4:

Designers of serial bus hardware need to ensure that there will be no metastability problems on SERDAT* as a result of sampling inputs from higher layers of serial bus management.

SUGGESTION 5.1:

Quand on conçoit des circuits possédant des modules de la couche de liaison de données, modifier les sorties vers les couches de haut niveau sur la transition C1 de SCLK, et échantillonner les entrées sur la carte venant des couches supérieures sur la transition S1. Spécifier les temps d'établissement et de maintien pour les couches supérieures en accord avec la transition S1 de SCLK ou, plutôt, prévoir une logique de synchronisation pour chaque entrée. Dans le dernier cas, spécifier en temps d'établissement par rapport à SCLK (front concerné) qui garantisse que le nouvel état d'une entrée sera reconnu et agira en conséquence.

5.1 Notations pour les diagrammes d'état

Ce chapitre comprend plusieurs diagrammes d'état qui définissent le comportement des modules du bus série. Lors de l'étude de ces diagrammes, il convient de garder à l'esprit les points suivants:

- 1) Quand sur une transition d'état apparaissent deux conditions ou plus, séparées par des virgules, la transition est franchie si toutes les conditions sont vraies. Ainsi, la notation "Dernier bit, XONE vrai" doit être lue "Dernier bit ET XONE vrai".
- 2) La condition d'entrée "RESET* bas" (qui logiquement débute chaque diagramme d'état) signifie que le module passe à l'état initial à partir de n'importe quel état, quand RESET* est au niveau bas.
- 3) Une boucle d'attente, sans condition explicitée, signifie que le module reste dans l'état pour toutes les conditions autres que celles indiquées sur les transitions de sortie de l'état.

5.2 Module EMETTEUR D'EN-TETE

Les couches supérieures utilisent un module EMETTEUR D'EN-TETE pour déclencher la transmission d'une trame sur le bus série. L'EMETTEUR D'EN-TETE envoie seulement la partie en-tête de la trame: le reste de la trame est envoyé par les autres modules RECEPTEURS D'EN-TETE, EMETTEURS DE DONNEES et RECEPTEURS DE DONNEES. Au fond, l'en-tête émis par ce module sert à sélectionner un ou plusieurs RECEPTEURS D'EN-TETE sur le bus série.

REGLE 5.1:

Un EMETTEUR D'EN-TETE DOIT toujours être apparié avec un GESTIONNAIRE DE TRAME, formant ainsi un groupe gestionnaire.

OBSERVATION 5.5:

Bien qu'un groupe gestionnaire occupe souvent la même carte que d'autres groupes de modules, ces groupes dialoguent seulement par la couche physique. Ainsi, le gestionnaire déclenche des trames concernant d'autres groupes sur la même carte, exactement de la même manière que pour des groupes hors de la carte.

SUGGESTION 5.1:

When designing circuits that implement Link Layer modules, change outputs to higher layers on the C1 transition of SCLK, and sample on-board inputs from higher layers on the S1 transition. Specify set-up and hold times for higher layers with respect to the S1 transition of SCLK or, preferably, provide synchronization logic for each input. In the latter case, specify a set-up time to (the relevant edge of) SCLK, which will guarantee that the new state of an input will be recognized and acted upon.

5.1 State diagram notation

This chapter includes a number of state diagrams that define serial bus modules in terms of their behaviour. The following points should be kept in mind when considering these diagrams:

- 1) When two or more conditions are listed on a state transition, separated by commas, the state transition is taken if all the conditions are True. Thus, the notation "Last Bit, XONE True" should be read "Last Bit AND XONE True".
- 2) The open-ended entry condition "RESET* Low" (which logically begins each state diagram) means that the module comes to this initial state from any state, when RESET* is Low.
- 3) A remain-in-state loop, with no condition labelled, means that the module remains in the state in all conditions other than those shown on the transitions out of the state.

5.2 HEADER SENDER module

Higher layers use a HEADER SENDER module to initiate the transmission of a frame on the serial bus. The HEADER SENDER sends only the Header portion of a frame: the remainder of the frame is sent by other modules including HEADER RECEIVERS, DATA SENDERS and DATA RECEIVERS. Basically, the Header sent by this module serves to select one or more HEADER RECEIVERS on the serial bus.

RULE 5.1:

A HEADER SENDER MUST always be paired with a FRAME MONITOR, thus making a Controller group.

OBSERVATION 5.5:

Even though a Controller group often occupies the same board with other module groups, these groups only communicate with each other via the Physical Layer. Thus, the Controller initiates frames involving other groups on the same board in exactly the same way that it does with off-board groups.

5.2.1 Interface avec la couche physique

Comme montré sur la figure 5-1, ci-dessous, l'interface avec la couche physique d'un EMETTEUR D'EN-TETE comprend les signaux SCLK, RONE, RSTART, XONE et XSTART.

5.2.2 Interface avec la couche de liaison de données

Comme montré sur la figure 5-1, l'interface avec la couche de liaison de données d'un EMETTEUR D'EN-TETE comprend les signaux suivants:

FRAME IN PROGRESS Entrée venant du GESTIONNAIRE DE TRAME apparié, indiquant si une trame est en cours de transmission sur le bus série.

SELECT Sortie vers le GESTIONNAIRE DE TRAME apparié indiquant que cet EMETTEUR D'EN-TETE a gagné l'arbitrage du bus série et déclenché la trame actuelle. Il s'ensuit que le GESTIONNAIRE DE TRAME renvoie le résultat de la transmission de trame vers la logique des couches supérieures.

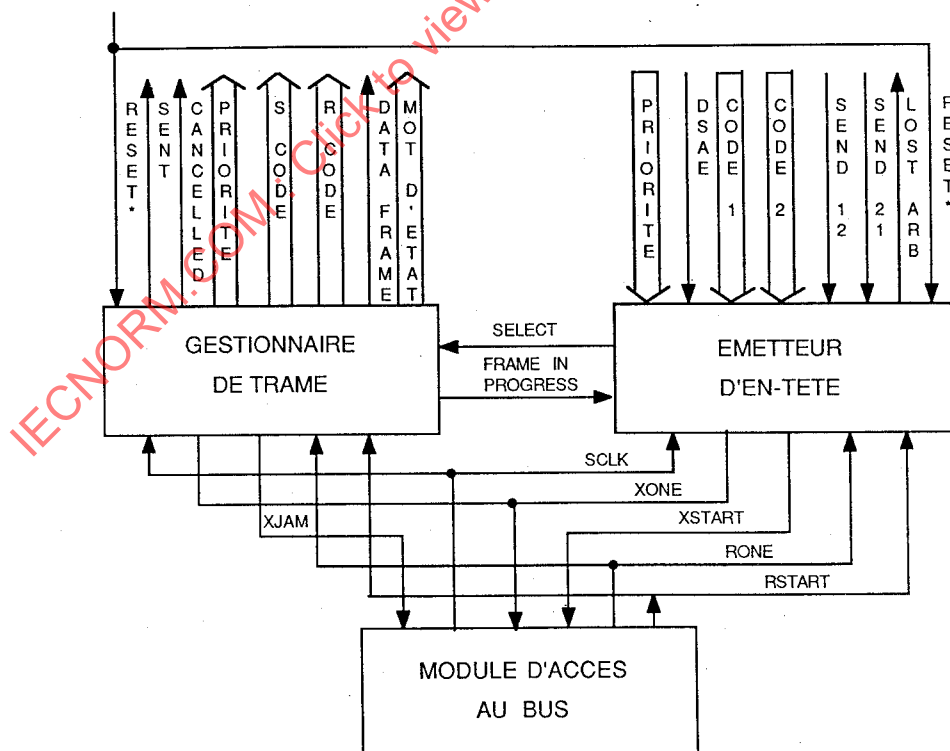


Figure 5-1 - EMETTEUR D'EN-TETE avec GESTIONNAIRE DE TRAME

5.2.1 Physical Layer interface

As shown in Figure 5-1, below, the Physical Layer interface of a HEADER SENDER includes the SCLK, RONE, RSTART, XONE and XSTART signals.

5.2.2 Link Layer interface

As shown in Figure 5-1, the Link Layer interface of a HEADER SENDER includes the following signals:

FRAME IN PROGRESS An input from the paired FRAME MONITOR, indicating whether a frame is currently in progress on the serial bus.

SELECT An output to a paired FRAME MONITOR, indicating that this HEADER SENDER has won the serial bus arbitration and initiated the current frame. This causes the FRAME MONITOR to report the result of frame transmission to higher-layer logic.

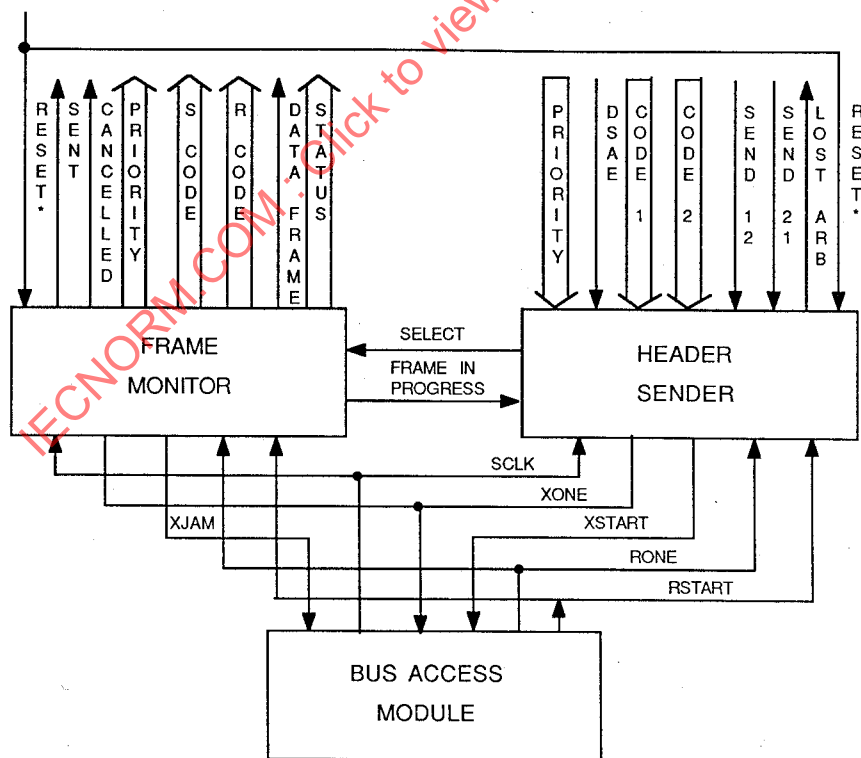


Figure 5-1 - HEADER SENDER with FRAME MONITOR

5.2.3 Interface avec les couches de niveau supérieur

Comme montré sur la figure 5-1, page 112, l'interface avec les couches supérieures de l'EMETTEUR D'EN-TETE comprend les signaux suivants:

RESET*	Quand ce signal est positionné, l'EMETTEUR D'EN-TETE est initialisé comme il est décrit dans le paragraphe suivant.
Port de priorité	Ensemble des lignes d'entrée et des signaux de contrôle associés au moyen desquels la valeur sur 3 bits de la priorité de trame pour la ou les transmissions des trames suivantes est chargée dans l'EMETTEUR D'EN-TETE. Ce port peut consister en des lignes d'entrée dédiées ou en un interface de bus avec un registre de priorité interne.
Ports de code 1 et 2	Deux ensembles semblables de signaux au moyen desquels deux adresses de 10 bits sont chargées pour les transmissions des trames suivantes.
Port DSAE	Ensemble de signaux semblables au moyen desquels le bit de validation de l'arbitrage de l'EMETTEUR DE DONNEES pour les transmissions de trames suivantes est chargé dans l'EMETTEUR D'EN-TETE.

AUTORISATION 5.1:

Les ports priorité, Code 1, Code 2 et DSAE PEUVENT être constitués d'entrées directes, ou de registres pouvant être chargés à partir de la même interface de bus, qui PEUVENT être reliés à un processeur sur la carte ou à l'interface de bus du système sur la carte.

SEND 12, SEND 21	Entrées de commande qui entraînent l'envoi par l'EMETTEUR D'EN-TETE d'une trame d'en-tête. SEND 12 implique l'envoi par l'EMETTEUR D'EN-TETE de la valeur du port de code 1 comme adresse S de l'en-tête et la valeur du port de code 2 comme adresse R. SEND 21 implique l'envoi par l'EMETTEUR D'EN-TETE de la valeur du port de code 2 comme adresse S et la valeur du port de code 1 comme adresse R.
------------------	---

OBSERVATION 5.6:

Les concepteurs du matériel du bus série peuvent être tentés de réaliser cette fonction SEND par un bit dans un registre interne, ou implicitement en chargeant une trame d'en-tête à transmettre. Cela restreint la généralité du circuit.

RECOMMANDATION 5.1:

Inclure une ou plusieurs entrées SEND dans les circuits d'EMETTEUR D'EN-TETE.

5.2.3 Higher Layer interface

As shown in Figure 5-1, page 113, the Higher Layer interface of a HEADER SENDER includes the following signals:

RESET* When this signal is asserted, the HEADER SENDER is initialized as described in the following paragraph.

Priority Port A set of input lines and associated control signals, through which the 3 bit Frame Priority value for subsequent frame transmission(s) is loaded into the HEADER SENDER. This port may consist of dedicated input lines, or a bus interface to an internal Priority register.

Code 1 and 2 Ports Two similar sets of signals, through which two 10-bit addresses for subsequent frame transmissions are loaded.

DSAE Port A similar set of signals, through which the DATA SENDER Arbitration Enable bit for subsequent frame transmissions is loaded into the HEADER SENDER.

PERMISSION 5.1:

The Priority, Code 1, Code 2 and DSAE Ports MAY be implemented as direct inputs, or as registers that can be loaded from the same bus interface, which MAY be connected to an on-board processor or on-board system bus interface.

SEND 12, SEND 21 Control inputs which cause the HEADER SENDER to send a Header subframe. SEND 12 makes the HEADER SENDER send the value from Code 1 Port as the S address in the Header, and the value from Code 2 Port as the R address. SEND 21 makes the HEADER SENDER send the value from Code 2 Port as the S address and the value from Code 1 Port as the R address.

OBSERVATION 5.6:

Implementors of serial bus hardware may be tempted to make this SEND function a bit in an internal register, or implied by loading a Header subframe to be transmitted. This restricts the applicability of the hardware circuit.

RECOMMENDATION 5.1:

Include one or more actual SEND inputs in HEADER SENDER circuits.

OBSERVATION 5.7:

En incluant une entrée SEND, un circuit devient applicable aussi bien à des cartes simples qu' "intelligentes". L'en-tête à transmettre peut être chargé à partir d'un processeur par le bus du système et la transmission véritable peut être déclenchée plus tard par un événement du signal de la carte.

LOST ARB Une sortie complémentaire à SELECT, indiquant que l'EMETTEUR D'EN-TETE a essayé d'envoyer une trame d'en-tête mais a perdu l'arbitrage pour l'utilisation du bus série au profit d'un autre EMETTEUR D'EN-TETE.

SUGGESTION 5.2:

Concevoir la logique des couches supérieures de sorte que lorsqu'un EMETTEUR D'EN-TETE perd l'arbitrage du bus série, la logique des couches supérieures augmente la valeur de la priorité pour les essais suivants (par exemple de un, peut être jusqu'à une limite supérieure définie)

OBSERVATION 5.8:

Cette stratégie, jointe à une réduction de la valeur de priorité quand une trame est annulée (voir paragraphe 5.6.3), aide à assurer que chaque EMETTEUR D'EN-TETE sur le bus série obtienne une part équitable de l'accès au bus.

OBSERVATION 5.9:

Les informations autres que LOST ARB, relatives aux résultats de l'envoi d'une trame, sont disponibles à partir des sorties du GESTIONNAIRE DE TRAME apparié.

5.2.4 Initialisation

Chaque fois que RESET* est au niveau bas, un EMETTEUR D'EN-TETE s'initialise de la façon suivante:

- 1) Il met sa sortie LOST ARB à l'état faux.
- 2) Il met XONE et XSTART à l'état faux.
- 3) Il rejette tout en-tête qui attendrait la transmission lorsque RESET* est passé au niveau bas.
- 4) Si l'un des ports de priorité, de code 1, de code 2 ou d'arbitrage de l'émetteur est constitué d'un registre chargeable, il met ces registres à zéro.

Quand RESET* passe au niveau haut, l'EMETTEUR D'EN-TETE entre dans "l'état repos".

5.2.5 Fonctionnement**REGLE 5.2:**

Après avoir effectué la procédure d'initialisation ci-dessus, l'EMETTEUR D'EN-TETE DOIT suivre le fonctionnement défini par le diagramme d'état de la figure 5-2, page 120, et par la description suivante.

OBSERVATION 5.7:

By including a SEND input, a circuit becomes applicable to "dumb" boards as well as "intelligent" ones. The Header to be transmitted can be loaded from a processor via the system bus and its actual transmission can be triggered later by an on-board signal event.

LOST ARB

An alternative output to SELECT, indicating that the HEADER SENDER has tried to send a Header subframe, but has lost the arbitration for use of the serial bus to another HEADER SENDER.

SUGGESTION 5.2:

Design higher-layer logic so that when a HEADER SENDER loses the serial bus arbitration, the higher-layer logic increases the Priority value for the subsequent retry (e.g. by one, perhaps up to a defined upper limit).

OBSERVATION 5.8:

This strategy, plus reduction of the Priority value when a frame is Cancelled (see Paragraph 5.6.3), helps ensure that each HEADER SENDER on the serial bus gets a fair share of access to the bus.

OBSERVATION 5.9:

Information other than LOST ARB, about the results of sending a frame, is available from the outputs of the paired FRAME MONITOR.

5.2.4 Initialization

Whenever RESET* is Low, a HEADER SENDER initializes itself as follows:

- 1) It makes its LOST ARB output False.
- 2) It makes XONE and XSTART False.
- 3) It discards any Header which was awaiting transmission when RESET* went Low.
- 4) If any of the Priority, Code 1, Code 2 or Sender Arb Ports are implemented as loadable registers, it clears those registers to zero.

When RESET* goes High, a HEADER SENDER enters "idle state".

5.2.5 Operation**RULE 5.2:**

After completing the above initialization procedure, a HEADER SENDER MUST operate as defined by the state diagram in Figure 5-2, page 121, and the following narrative description.

Partant de l'état repos, l'EMETTEUR D'EN-TETE procède de la façon suivante:

- 1) Il échantillonne SEND 12 et SEND 21 sur chaque front S1 de SCLK, surveille ensuite RONE, échantillonne FRAME IN PROGRESS sur chaque front S2 de SCLK et contrôle ensuite RSTART.

SI il échantillonne SEND 12 ou SEND 21 vrai sur un front S1, puis trouve RONE faux, puis échantillonne FRAME IN PROGRESS faux sur le front S2 suivant et, enfin, trouve RSTART faux (c'est-à-dire s'il échantillonne un bit_zéro).

ALORS il met XSTART à l'état vrai avant le front C2 du cycle de bit suivant pour générer un bit_départ. L'EMETTEUR D'EN-TETE passe ensuite à l'étape 2 ci-dessous.

SINON

SI il échantillonne SEND 12 ou SEND 21 vrai sur un front S1, puis trouve RONE faux, puis échantillonne FRAME IN PROGRESS faux sur le front S2 suivant et enfin trouve RSTART vrai (c'est-à-dire si un autre EMETTEUR D'EN-TETE envoie un bit_départ. Cela ne peut se produire que si la logique sur la carte met SEND 12 ou SEND 21 à l'état vrai pendant le cycle précédent de SCLK).

ALORS il attend simplement un cycle de SCLK et retourne à l'état de repos où il surveille ses entrées comme décrit ci-dessus. A la fin de la trame actuelle, il pourra à nouveau passer à l'étape 2.

SINON (SEND 12 et SEND 21 sont faux, ou FRAME IN PROGRESS est vrai, ou RONE est vrai) l'EMETTEUR D'EN-TETE reste dans l'état de repos.

- 2) Avant le front C1 suivant de SCLK, l'EMETTEUR D'EN-TETE place le bit le plus significatif de sa priorité de trame sur XONE. Avant le front C2 suivant, il met RSTART à l'état faux.

OBSERVATION 5.10:

D'autres EMETTEURS D'EN-TETE peuvent aussi placer le bit le plus significatif de leur priorité de trame sur le bus série dans la même tranche de temps.

- 3) Après le front S1 suivant SCLK, l'EMETTEUR D'EN-TETE contrôle RONE.

SI il met XONE faux avant C1, et qu'il trouve RONE vrai après le front S1,

ALORS cet EMETTEUR D'EN-TETE a perdu l'arbitrage pour l'utilisation du bus série. Dans ce cas, il met sa sortie LOST ARB vraie pendant un cycle SCLK et retourne à l'état de repos.

- 4) Le processus décrit aux étapes 2 et 3 est répété pour chacun des bits restants dans l'en-tête. Cela inclut les deux bits restants du champ de priorité de trame, les adresses R et S de 10 bits chacune et le "bit de validation d'arbitrage de l'émetteur".

Starting from idle state, a HEADER SENDER proceeds as follows:

- 1) It samples SEND 12 and SEND 21 on each S1 edge of SCLK, monitors RONE thereafter, samples FRAME IN PROGRESS on each S2 edge of SCLK and monitors RSTART thereafter.

IF it samples SEND 12 or SEND 21 True on an S1 edge, then monitors RONE False, then samples FRAME IN PROGRESS False on the following S2 edge, and then monitors RSTART False (i.e. if it samples a zero_bit).

THEN it makes XSTART True before the C2 edge of the next bit cycle, to generate a start_bit. The HEADER SENDER then proceeds to step 2 below.

ELSE

IF it samples SEND 12 or SEND 21 True on an S1 edge, then monitors RONE False, then samples FRAME IN PROGRESS False on the following S2 edge, and then monitors RSTART True (i.e. if another HEADER SENDER is sending a start_bit; this can occur only if on-board logic made SEND 12 or SEND 21 True during the previous SCLK cycle).

THEN it simply delays one cycle of SCLK and returns to idle state sampling as described above. At the end of the current frame, it will be able to proceed to step 2.

ELSE (SEND 12 and SEND 21 are False, or FRAME IN PROGRESS is True, or RONE is True) the HEADER SENDER remains in idle state.

- 2) Before the next C1 edge of SCLK, the HEADER SENDER places the most significant bit of its Frame Priority value on XONE. Before the next C2 edge it makes RSTART False.

OBSERVATION 5.10:

Other HEADER SENDERS may also be placing the MSB of their Frame Priority values on the serial bus in this same bit time.

- 3) After the following S1 edge of SCLK, the HEADER SENDER monitors RONE.

IF it made XONE False before C1, and it monitors RONE True after the S1 edge,

THEN this HEADER SENDER has lost the arbitration for use of the serial bus. In this case, it makes its LOST ARB output True for one SCLK cycle and returns to idle state.

- 4) The process described in steps 2 and 3 is repeated for each of the remaining bits in the Header. This includes the remaining two bits of the Frame Priority field, the S and R addresses which are 10 bits each and the "sender arbitration enable bit".

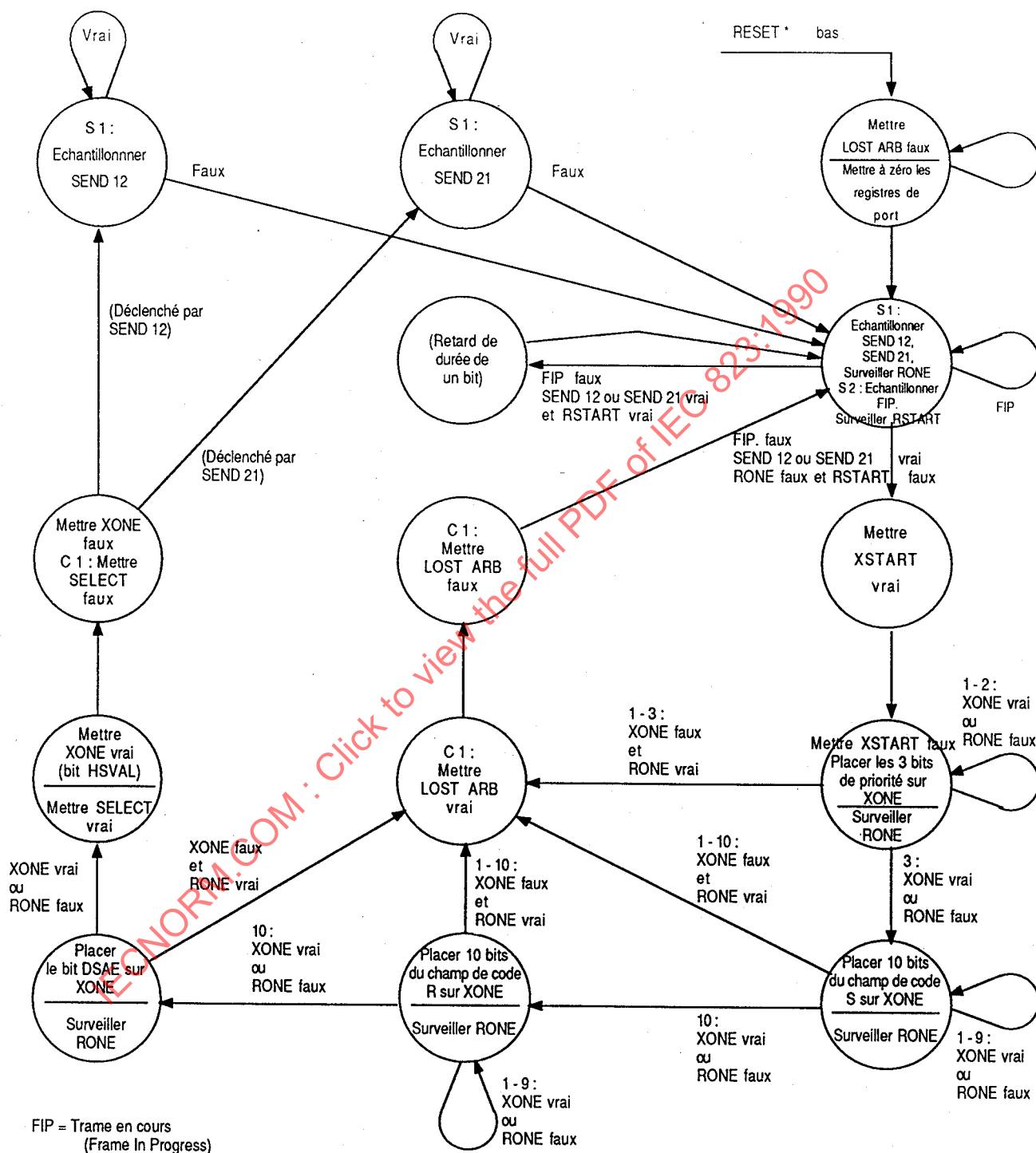


Figure 5-2 - Diagramme d'état de l'EMETTEUR D'EN-TETE

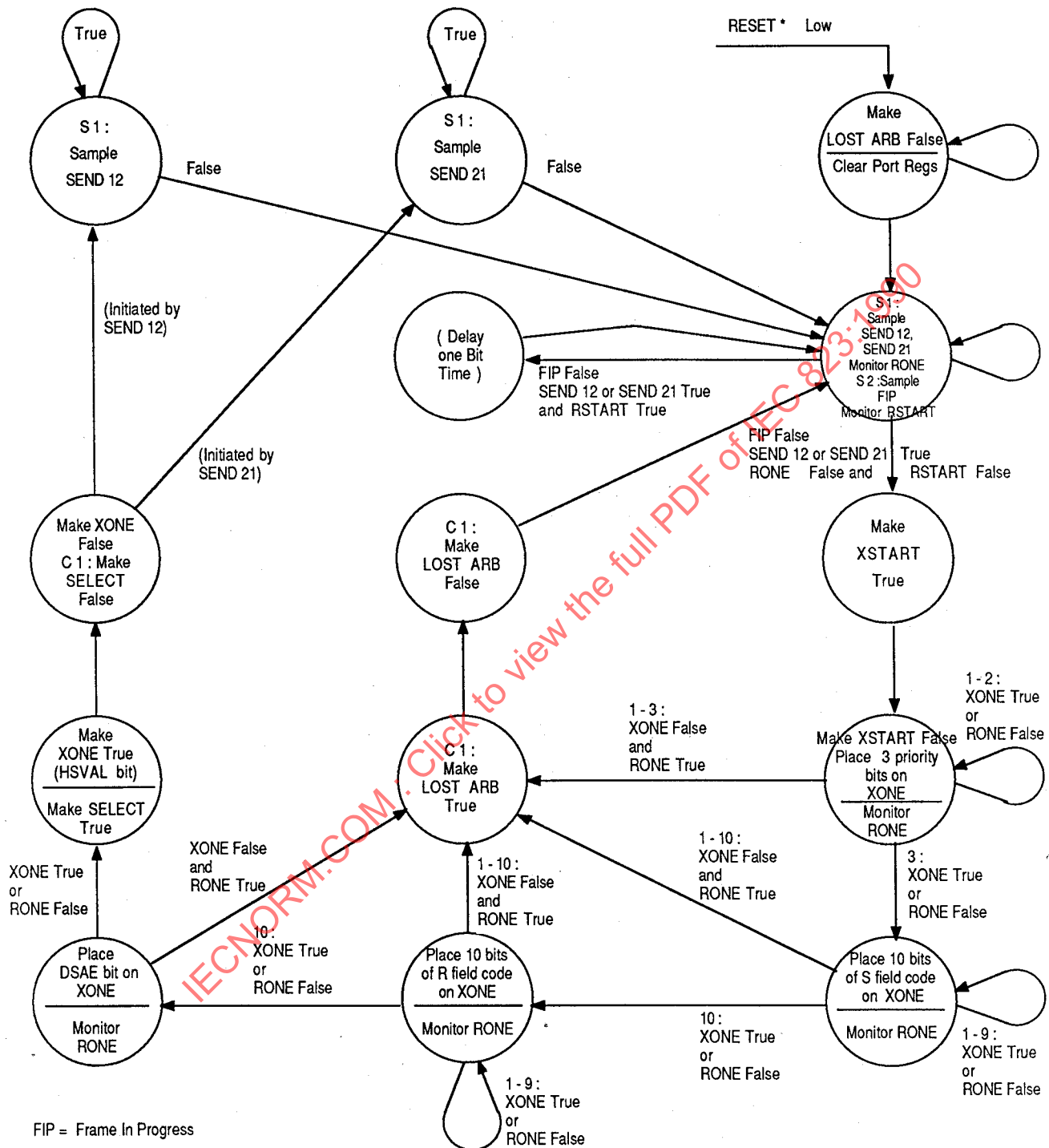


Figure 5-2 - HEADER SENDER state diagram

- 5) Avant le front C1 suivant, l'EMETTEUR D'EN-TETE positionne XONE à l'état vrai pour le bit HSVAL, et sur C1 positionne aussi sa sortie SELECT à l'état vrai. SELECT à l'état vrai indique que cet EMETTEUR D'EN-TETE a gagné l'arbitrage du bus série, a envoyé son en-tête avec succès et fait que l'EMETTEUR DE TRAME apparié reporte les résultats de toute la transmission de la trame vers la logique de la couche supérieure sur la carte.

OBSERVATION 5.11:

Le bit HSVAL empêche la possibilité que tous les EMETTEURS D'EN-TETE se retirent de l'arbitrage à cause du bruit sur le bus. Si le bit HSVAL d'un message quelconque est un bit_zéro, la trame est ignorée par tous les RECEPTEURS D'EN-TETE.

- 6) Après avoir envoyé HSVAL et positionné SELECT à l'état vrai, avant le front C1 suivant, l'EMETTEUR D'EN-TETE positionne XONE à l'état faux, et sur ce front il met SELECT à l'état faux. Sur les fronts suivants S1, il échantillonne SEND 12 et SEND 21, jusqu'à ce que le signal qui a déclenché l'opération (étape 1) soit mis à l'état faux par la logique de la couche supérieure. Il retourne ensuite à l'état repos.

Noter que cette méthode d'arbitrage garantit que l'en-tête d'au moins un EMETTEUR D'EN-TETE est toujours envoyé correctement (empêchant une erreur de "désynchronisation" due à un bruit extérieur induit). Si l'on considère les en-têtes concurrents comme des nombres binaires, celui qui est envoyé avec succès est celui qui a la plus grande valeur binaire.

Mais si deux EMETTEURS D'EN-TETE ou plus envoient exactement le même en-tête, ils n'auront pas connaissance de la présence de l'autre et chacun continuera comme s'il avait envoyé l'en-tête seul. Cela est parfait pour certaines trames. Ce n'est pas approprié pour d'autres trames (par exemple celles qui positionnent un "sémaphore").

REGLE 5.3:

Quand la fonction d'une trame nécessite que seul un EMETTEUR D'EN-TETE soit autorisé à envoyer l'en-tête à un moment donné, les couches supérieures du gestionnaires du bus série DOIVENT assurer que la valeur de l'en-tête est différente de celle envoyée par n'importe quel autre EMETTEUR D'EN-TETE.

OBSERVATION 5.12:

Si cette règle est vérifiée, l'arbitrage normal du bus série assure que tous les EMETTEURS D'EN-TETE sauf un se retireront du bus avant que l'en-tête soit complètement envoyé.

5.3 Module RECEPTEUR D'EN-TETE

Un RECEPTEUR D'EN-TETE peut être seul ou associé à un EMETTEUR DE DONNEES et/ou un RECEPTEUR DE DONNEES. Sa fonction première est de comparer les adresses S et R dans chaque en-tête sur le bus série par rapport à son adresse, et de signaler à ses modules appariés et/ou aux couches supérieures s'il trouve une coïncidence.

- 5) Before the next C1 edge, the HEADER SENDER makes XONE True for the HSVAL bit, and at C1 also makes its SELECT output True. SELECT True indicates that this HEADER SENDER has won the serial bus arbitration and sent its Header successfully, and makes the paired FRAME MONITOR report the results of the overall frame transmission to on-board higher-layer logic.

OBSERVATION 5.11:

The HSVAL bit guards against the possibility that all HEADER SENDERS dropped out of the arbitration due to noise on the bus. If the HSVAL bit of any message frame is a zero_bit, the frame is ignored by all HEADER RECEIVERS.

- 6) After sending HSVAL and making SELECT True, before the next C1 edge the HEADER SENDER makes XONE False, and on that edge it makes SELECT False. On subsequent S1 edges it samples SEND 12 and SEND 21, until the signal which initiated the operation (step 1) is made False by higher-layer logic. It then returns to idle state.

Note that this arbitration method guarantees that the Header from at least one HEADER SENDER is always sent correctly (barring an "all drop out" error due to externally induced noise). If one considers the contending Headers as binary numbers, the one that is successfully sent is the one with the highest binary value.

But if two or more HEADER SENDERS set out to send exactly the same Header, they will not know of each other's presence, and each will proceed as if it had sent the Header alone. This is fine for some frames. It is not appropriate for other frames (e.g. those which set a "semaphore").

RULE 5.3:

When the function of a frame requires that only one HEADER SENDER be allowed to send the Header at a time, higher layers of serial bus management MUST ensure that the value of the Header is different from that sent by any other HEADER SENDER.

OBSERVATION 5.12:

If this rule is met, the normal serial bus arbitration ensures that all but one HEADER SENDER will retire from the bus before the Header completes.

5.3 HEADER RECEIVER module

A HEADER RECEIVER may stand alone or may be paired with a DATA SENDER and/or a DATA RECEIVER. Its primary function is to compare the S and R addresses in each Header on the serial bus against its address, and signal its paired modules and/or higher layers if it finds a match.

5.3.1 Interface avec la couche physique

Comme indiqué dans la figure 5-3, page 126, l'interface avec la couche physique d'un RECEPTEUR D'EN-TETE inclut les signaux SCLK, RONE, RSTART et XONE.

5.3.2 Interface avec la couche de liaison de données

Comme indiqué dans la figure 5-3, l'interface avec la couche de liaison d'un RECEPTEUR D'EN-TETE inclut les signaux suivants:

S, R SELECT Deux sorties vers un EMETTEUR DE DONNEES et un RECEPTEUR DE DONNEES appariés, respectivement, qui indiquent que le RECEPTEUR D'EN-TETE a détecté un en-tête sur le bus série qui inclut son adresse dans les champs S et R, respectivement. Ces sorties sont positionnées à l'état vrai pour une période SCLK.

DSAE Une sortie vers un EMETTEUR DE DONNEES apparié, qui reflète l'état du bit DSAE dans un en-tête.

5.3.3 Interface avec les couches supérieures

Comme indiqué dans la figure 5-3, l'interface avec la couche supérieure d'un RECEPTEUR D'EN-TETE inclut les signaux suivants:

RESET* Quand ce signal est validé, le RECEPTEUR D'EN-TETE est initialisé comme décrit dans la section suivante.

Port de code Un ensemble de lignes d'entrée parallèles et de signaux de commande associés, à travers lequel une valeur d'une adresse de 10 bits peut être chargée dans le RECEPTEUR D'EN-TETE.

AUTORISATION 5.2:

Le port de code PEUT être réalisé sous forme d'entrées dédiées, ou comme un interface de bus vers un registre d'adresse interne.

ENABLE S, R Deux entrées qui indiquent s'il faut accepter une trame avec cette adresse de module dans le champ S ou R, respectivement. Si le RECEPTEUR D'EN-TETE détecte une coïncidence d'adresse S (ou R), et que l'entrée ENABLE S (ENABLE R) est à l'état faux, il annule la trame.

S, R STROBE Deux sorties qui représentent la même information que SELECT S et SELECT R, mais sont positionnées à l'état vrai à la fin de la trame et seulement si la trame est terminée avec succès.

5.3.1 Physical Layer Interface

As shown in Figure 5-3, page 127, the Physical Layer interface of a HEADER RECEIVER includes the SCLK, RONE, RSTART, and XONE signals.

5.3.2 Link Layer Interface

As shown in Figure 5-3, the Link Layer interface of a HEADER RECEIVER includes the following signals:

S, R SELECT Two outputs to a paired DATA SENDER and DATA RECEIVER, respectively, which indicate that the HEADER RECEIVER has detected a Header on the serial bus that includes its address in the S or R field, respectively. These outputs are made True for one SCLK period.

DSAE An output to a paired DATA SENDER, which reflects the state of the DSAE bit in a Header.

5.3.3 Higher Layer Interface

As shown in Figure 5-3, the Higher Layer interface of a HEADER RECEIVER includes the following signals:

RESET* When this signal is asserted, the HEADER RECEIVER is initialized as described in Paragraph 5.3.4.

Code Port A set of parallel input lines and associated control signals, through which a 10-bit address value can be loaded into the HEADER RECEIVER.

PERMISSION 5.2:

The Code Port MAY be implemented as dedicated inputs, or as a bus interface to an internal address register.

ENABLE S, R Two inputs which indicate whether to accept a frame with this module's address in the S or R field, respectively. If the HEADER RECEIVER matches the S (or R) address, and the ENABLE S (ENABLE R) input is False, it cancels the frame.

S, R STROBE Two outputs which indicate the same information as S SELECT and R SELECT, but are made True at the end of the frame only if the frame is completely successful.

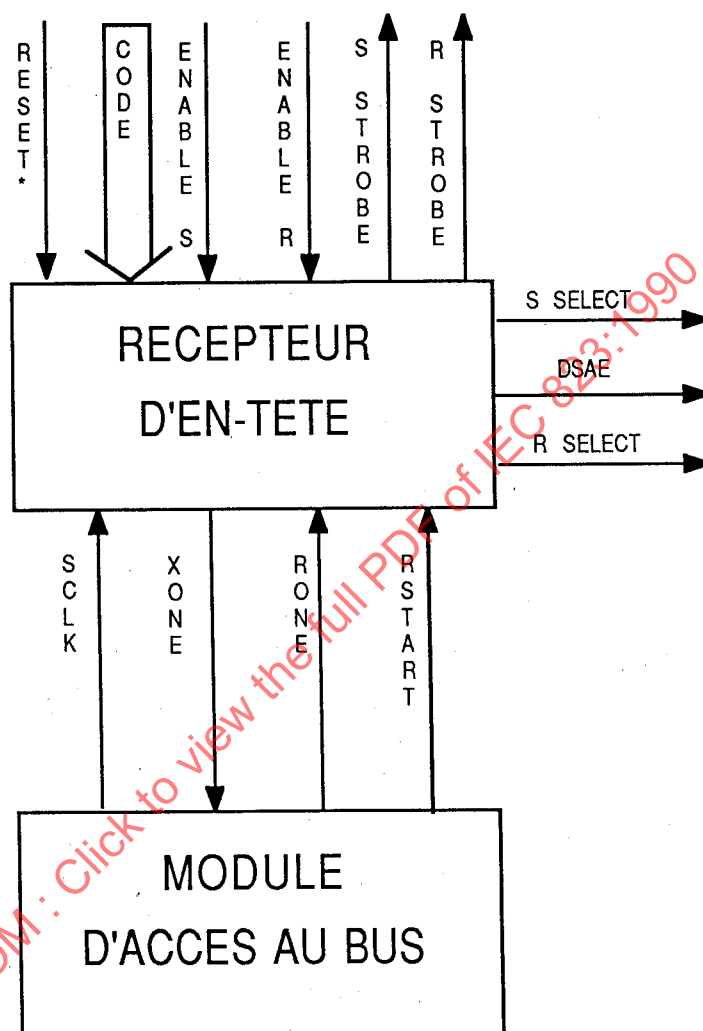


Figure 5-3 - Signaux utilisés par un RECEPTEUR D'EN-TETE

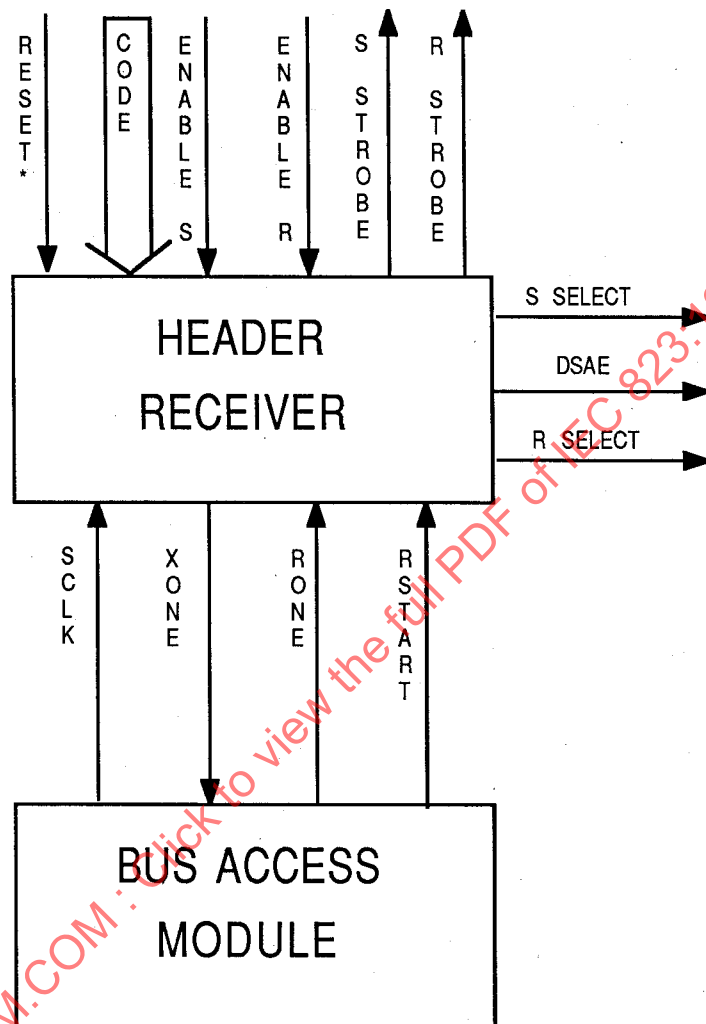


Figure 5-3 - Signals used by a HEADER RECEIVER

5.3.4 Initialisation

Chaque fois que RESET* est au niveau bas, un RECEPTEUR D'EN-TETE s'initialise comme suit:

- 1) Il positionne ses sorties S SELECT, R SELECT, S STROBE et R STROBE à l'état faux.
- 2) Il positionne XONE à l'état faux, et ignore RONE et RSTART.
- 3) Si le port de code est réalisé comme un registre chargeable, il met le registre à zéro.

Quand RESET* passe au niveau haut, le RECEPTEUR D'EN-TETE entre dans un "état de repos".

5.3.5 Programmation de l'adresse

Si l'adresse est configurée comme un registre chargeable, le RECEPTEUR D'EN-TETE doit être programmé avec la valeur souhaitée après initialisation. Pour les besoins de cette norme, cela est considéré comme une partie du démarrage du système avant qu'une trame ne soit émise sur le bus série. Il est autorisé de reprogrammer un RECEPTEUR D'EN-TETE avec une nouvelle adresse pendant que le bus série est opérationnel, mais cela n'est pas couvert par cette norme.

5.3.6 Fonctionnement

REGLE 5.4:

Après avoir terminé la procédure d'initialisation ci-dessus, un RECEPTEUR D'EN-TETE DOIT opérer comme défini par le diagramme d'état de la figure 5-4, page 130, et la description narrative suivante.

Partant d'un état repos, un RECEPTEUR D'EN-TETE procède comme suit:

- 1) Il surveille continuellement RSTART, jusqu'à ce qu'il détecte RSTART à l'état vrai après un front S2 de SCLK. Cela indique un bit départ sur le bus série. Sur le premier front C1 après être retourné des étapes 15 ou 22 ci-après, un RECEPTEUR D'EN-TETE positionne ses sorties S STROBE et R STROBE à l'état faux.
- 2) Dès que le RECEPTEUR D'EN-TETE a détecté un bit-départ, il laisse passer ensuite trois périodes d'un bit sur SCLK, ignorant de cette manière le champ de priorité de la trame.
- 3) Le RECEPTEUR D'EN-TETE surveille ensuite les 10 bits suivants sur RONE, après chaque front S1 de SCLK. Il compare chacun de ces bits aux bits correspondants de son adresse du port de code. Si tous les 10 bits sont égaux, il positionne un signal interne appelé "S MATCH" à l'état vrai, sinon il positionne S MATCH à l'état faux.

5.3.4 Initialization

Whenever RESET* is Low, a HEADER RECEIVER initializes itself as follows:

- 1) It makes its S SELECT, R SELECT, S STROBE and R STROBE outputs False.
- 2) It makes XONE False, and ignores RONE and RSTART.
- 3) If the Code Port is implemented as a loadable register, it clears the register to zero.

When RESET* goes high, the HEADER RECEIVER enters "idle state".

5.3.5 Programming the address

If the address is implemented as a loadable register, the HEADER RECEIVER has to be programmed with the desired value after initialization. For the purposes of this standard, this is considered part of system set-up before any frames are sent on the serial bus. Reprogramming a HEADER RECEIVER with a new address, while the serial bus is in operation, is allowed but not covered by this standard.

5.3.6 Operation

RULE 5.4:

After completing the above initialization procedure, a HEADER RECEIVER MUST operate as defined by the state diagram in Figure 5-4, page 131, and the following narrative description.

Starting from idle state, a HEADER RECEIVER proceeds as follows:

- 1) It continuously monitors RSTART, until it monitors RSTART True after an S2 edge of SCLK. This indicates a start_bit on the serial bus. On the first C1 edge after returning from step 15 or 22 below, a HEADER RECEIVER makes its S STROBE and R STROBE outputs False.
- 2) Once the HEADER RECEIVER has detected a start_bit, it then counts off three bit-times on SCLK, thus ignoring the Frame Priority field of the frame.
- 3) The HEADER RECEIVER then monitors the next 10 bits on RONE, after each S1 edge of SCLK. It compares each to the corresponding bit of its address from Code Port. If all 10 bits are equal it makes an internal signal called S MATCH True; if not it makes S MATCH False.

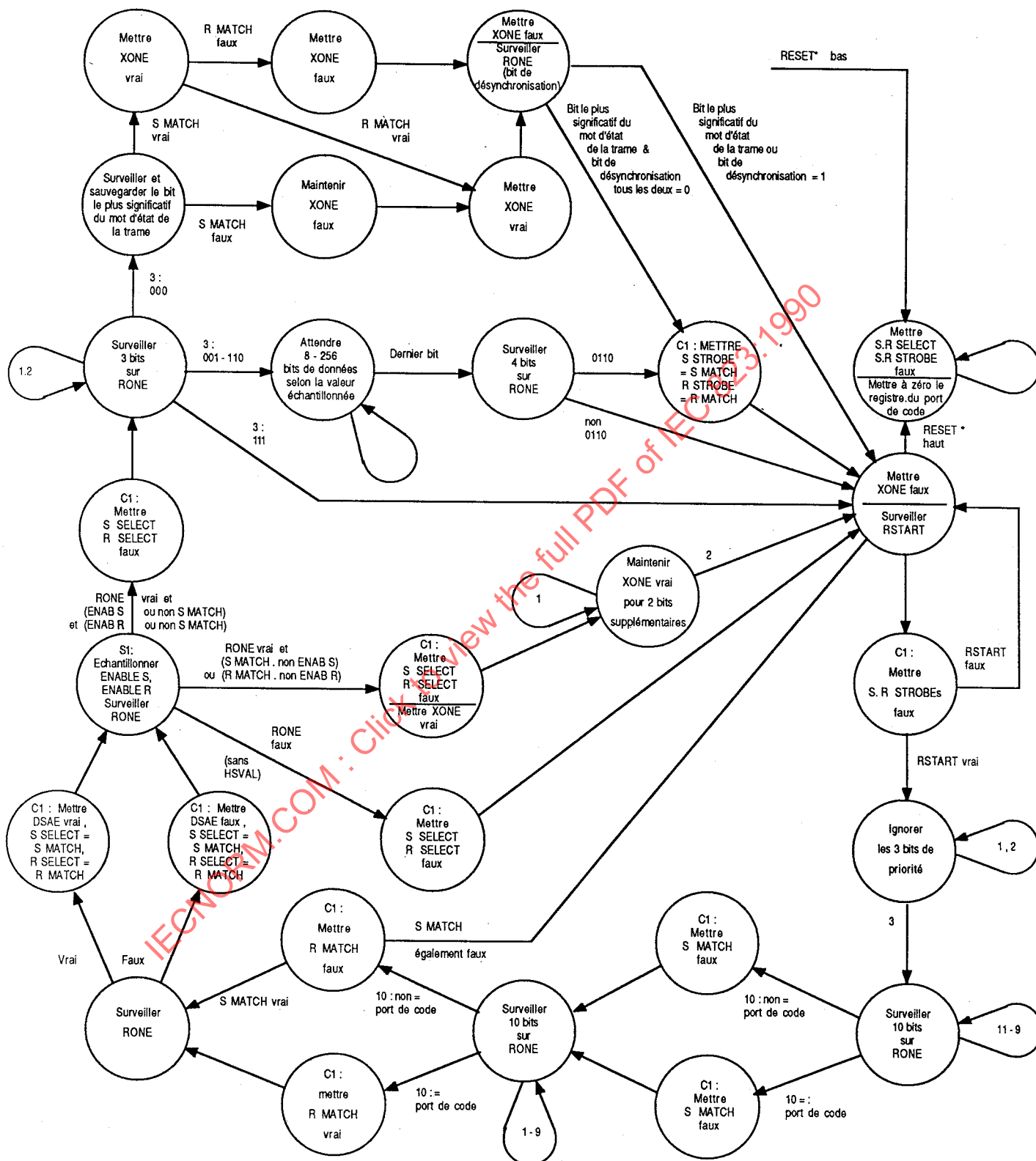


Figure 5-4 - Diagramme d'état du RECEPTEUR D'EN-TETE

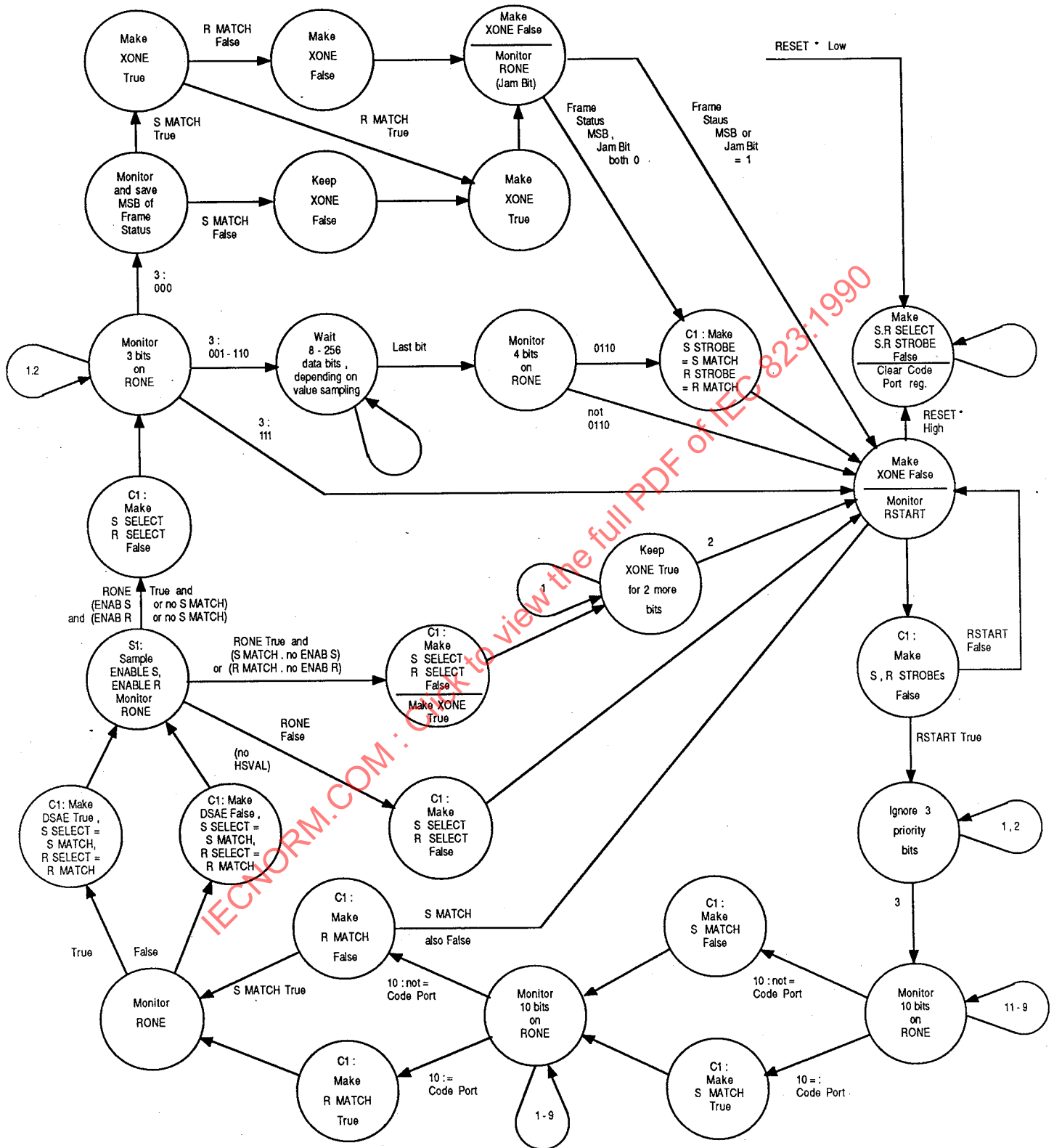


Figure 5-4 - HEADER RECEIVER state diagram

- 4) Le RECEPTEUR D'EN-TETE surveille ensuite les 10 bits suivants sur RONE comme à l'étape 3 excepté pour le positionnement à l'état vrai ou faux d'un signal interne appelé "R MATCH". Si S MATCH et R MATCH sont maintenant tous les deux à l'état faux, le RECEPTEUR D'EN-TETE retourne à l'état de repos (étape 1).
- 5) Il surveille ensuite le bit suivant sur RONE après le front S1 suivant de SCLK. C'est le bit "Validation de l'arbitrage de l'émetteur". Si RONE est à l'état vrai il positionne sa sortie DSAE à l'état vrai, sinon il positionne DSAE à l'état faux.
- 6) Sur le front C1 suivant de SCLK, si S MATCH est vrai, le RECEPTEUR D'EN-TETE positionne sa sortie S SELECT à l'état vrai. Sur le même front, si R MATCH est vrai, il positionne R SELECT à l'état vrai.
- 7) Sur le front S1 suivant de SCLK, le RECEPTEUR D'EN-TETE échantillonne ses entrées ENABLE S et ENABLE R. Il surveille ensuite le bit HSVAL sur RONE.

Selon l'état de HSVAL sur RONE, de S MATCH et R MATCH, et de l'échantillonnage de ENABLE S et ENABLE R, le RECEPTEUR D'EN-TETE procède comme suit:

HSVAL	S MATCH	R MATCH	ENAB S	ENAB R	Action
zéro	← les deux ne sont pas faux →		x	x	Etape 8
un	Vrai	x	Faux	x	Etape 9
un	x	Vrai	x	Faux	Etape 9
un	← n'importe quelle autre combinaison →				Etape 10

OBSERVATION 5.13:

Un "x" dans le tableau ci-dessus indique que ce paramètre est indifférent au RECEPTEUR D'EN-TETE dans ce cas.

- 8) (Le bit HSVAL était un bit_zéro.) Le RECEPTEUR D'EN-TETE positionne simplement S SELECT et R SELECT à l'état faux sur le front C1 suivant et retourne à l'état de repos.
- 9) (Dans ce cas le RECEPTEUR D'EN-TETE a une adresse qui coïncide avec l'adresse S de l'en-tête et a échantillonné l'entrée ENABLE S à l'état faux, ou il a une adresse qui coïncide avec l'adresse R et a échantillonné l'entrée ENABLE R à l'état faux.) Dans les deux cas, avant le front C1 suivant, il positionne XONE à l'état vrai, et sur le même front il positionne S SELECT et R SELECT à l'état faux. Il maintient XONE vrai durant les deux fronts C1 suivants. Cela "annule" la trame sur le bus série. Après le troisième front C1, il positionne XONE à l'état faux et retourne à l'état de repos.

- 4) The HEADER RECEIVER then monitors the next 10 bits on RONE as it did in step 3, except that it makes an internal signal called "R MATCH" True or False. If S MATCH and R MATCH are now both False, the HEADER RECEIVER returns to idle state (step 1).
- 5) It then monitors the next bit on RONE after the next S1 edge of SCLK. This is the "Sender Arb Enable" bit. If RONE is True it makes its DSAE output True, otherwise it makes DSAE False.
- 6) On the next C1 edge of SCLK, if S MATCH is True, the HEADER RECEIVER makes its S SELECT output True. On the same edge, if R MATCH is True it makes R SELECT True.
- 7) On the next S1 edge of SCLK, the HEADER RECEIVER samples its ENABLE S and ENABLE R inputs. It then monitors the (HSVAL) bit on RONE.

Depending on the HSVAL state of RONE, the state of S MATCH and R MATCH, and how it sampled ENABLE S and ENABLE R, the HEADER RECEIVER proceeds as follows:

HSVAL	S MATCH	R MATCH	ENAB S	ENAB R	Action
zero	← not both False →		x	x	Step 8
one	True	x	False	x	Step 9
one	x	True	x	False	Step 9
one	← any other combination →				Step 10

OBSERVATION 5.13:

An "x" in the table above indicates that this item does not matter to the HEADER RECEIVER in this case.

- 8) (The HSVAL bit was a zero_bit.) The HEADER RECEIVER simply makes S SELECT and R SELECT False on the next C1 edge and returns to idle state.
- 9) (In this case the HEADER RECEIVER has matched the S address in the Header and sampled the ENABLE S input False, or it has matched the R address and sampled the ENABLE R input False.) In either case, before the next C1 edge it makes XONE True, and on that edge it makes S SELECT and R SELECT False. It maintains XONE True through the following two C1 edges. This "cancels" the frame on the serial bus. After the third C1 edge, it makes XONE False and returns to idle state.

- 10) (Dans ce cas le RECEPTEUR D'EN-TETE a détecté la coïncidence de l'adresse S ou R, a positionné sa sortie S SELECT et/ou R SELECT à l'état vrai et a échantillonné l'entrée ENABLE correspondante à l'état vrai.) Il positionne maintenant S SELECT et R SELECT à l'état faux sur le front C1 suivant et surveille ensuite RONE après chacun des trois fronts S1 suivants sur SCLK. Cela est le code de type de trame.
- 11) Si le type de trame est 111, la trame a été annulée par un autre RECEPTEUR D'EN-TETE et ce module retourne à l'état de repos.
- 12) Si le code de type de trame est 000, le RECEPTEUR D'EN-TETE continue à l'étape 17. Autrement (le code de type de trame est 001-110), c'est une trame de transfert de données. Dans ce cas, le RECEPTEUR D'EN-TETE se comporte comme un module "de poursuite" dans lequel son signal de sortie S STROBE (ou R STROBE) indique un transfert de données réussi de (ou vers) un autre module avec la même adresse.
- 13) Le RECEPTEUR D'EN-TETE laisse passer le nombre nécessaire de cycles SCLK pour ignorer la sous-trame de données sur RONE. Il détermine le nombre de cycles de retard à partir du type de trame échantillonné, comme suit:

Type de trame	Nombre de SCLK
001	8
010	16
011	32
100	64
101	128
110	256

- 14) Après le nombre ci-dessus de cycles SCLK, le RECEPTEUR D'EN-TETE surveille RONE après chacun des quatre fronts suivants S1 de SCLK. Ces quatre bits représentent le champ de mot d'état de la trame et le bit de détection de désynchronisation.
- 15) Si les quatre bits sont exactement 0110, alors le transfert de données sur le bus série est réussi. Sur le front C1 suivant, le RECEPTEUR D'EN-TETE positionne la sortie S STROBE à l'état vrai si S MATCH est vrai, positionne la sortie R STROBE à l'état vrai si R MATCH est vrai, et retourne à l'état de repos.
- 16) Si les quatre bits sont différents de 0110, le transfert de données sur le bus série n'a pas réussi. Le RECEPTEUR D'EN-TETE n'active pas S ou R STROBE, mais retourne simplement à l'état de repos.
- 17) (Le type de trame était 000, et la trame ne contient pas de données.) Le RECEPTEUR D'EN-TETE surveille RONE pour le bit le plus significatif de champ du mot d'état de la trame après le front S1 suivant, et mémorise son mot d'état pour plus tard.
- 18) Avant le front C1 suivant, si S MATCH est à l'état vrai, le RECEPTEUR D'EN-TETE positionne XONE à l'état vrai pour indiquer sa présence. (Si S MATCH est à l'état faux, il maintient XONE à l'état faux.)
- 19) Avant le front C1 suivant, si R MATCH est à l'état vrai, le RECEPTEUR D'EN-TETE positionne XONE à l'état vrai pour indiquer sa présence. (Si R MATCH est à l'état faux, il positionne XONE à l'état faux.)

- 10) (In this case the HEADER RECEIVER has matched the S or the R address, set its S SELECT and/or its R SELECT output True and sampled the corresponding ENABLE input True.) It now makes S SELECT and R SELECT False on the next C1 edge, and then monitors RONE after each of the next three S1 edges of SCLK. This is the Frame Type code.
- 11) If the Frame Type is 111, the frame has been cancelled by another HEADER RECEIVER and this module returns to idle state.
- 12) If the Frame Type code is 000, the HEADER RECEIVER proceeds to step 17. Otherwise (the Frame Type code is 001-110), this is a Data Transfer frame. The HEADER RECEIVER in this case acts as a "tracker" module wherein its S STROBE (or R STROBE) output signals a successful data transfer from (or to) another module with the same address.
- 13) The HEADER RECEIVER counts off the necessary number of SCLK cycles to ignore the Data subframe on RONE. It determines the number of cycles to delay from the sampled Frame Type, as follows:

<i>Frame Type</i>	<i>SCLK count</i>
001	8
010	16
011	32
100	64
101	128
110	256

- 14) After the above number of SCLK cycles, the HEADER RECEIVER monitors RONE after each of the next four S1 edges of SCLK. These four bits are the Frame Status field and the Jam Detect bit.
- 15) If the four bits are exactly 0110, then the serial bus data transfer is successful. On the following C1 edge the HEADER RECEIVER makes the S STROBE output True if S MATCH is True, makes the R STROBE output True if R MATCH is True, and returns to idle state.
- 16) If the four bits are anything other than 0110, the serial bus data transfer was not successful. The HEADER RECEIVER does not signal on S or R STROBE, but simply returns to idle state.
- 17) (The Frame Type was 000, and the frame contains no data.) The HEADER RECEIVER monitors RONE for the MSB of the Frame Status field after the next S1 edge, and stores its state for later.
- 18) Before the next C1 edge, if S MATCH is True, the HEADER RECEIVER makes XONE True to show that it is present. (If S MATCH is False, it keeps XONE False.)
- 19) Before the next C1 edge, if R MATCH is True, the HEADER RECEIVER makes XONE True to show that it is present. (If R MATCH is False, it makes XONE False.)

- 20) Avant le front C1 suivant, le RECEPTEUR D'EN-TETE positionne XONE à l'état faux et puis surveille RONE après le front S1 suivant. C'est le bit de détection de désynchronisation.
- 21) Si le bit de détection de désynchronisation est un bit_un ou le bit le plus significatif précédemment mémorisé du champ de mot d'état était un bit_un, le RECEPTEUR D'EN-TETE ne positionne pas ses sorties S STROBE ou R STROBE à l'état vrai, mais retourne simplement à l'état de repos.
- 22) (Le bit de détection de désynchronisation et le bit le plus significatif précédemment mémorisé du mot d'état de la trame sont tous les deux des bit_zéro.) Si S MATCH est à l'état vrai, le RECEPTEUR D'EN-TETE positionne sa sortie S STROBE à l'état vrai sur le front C1 suivant. Sur le même front, si R MATCH est à l'état vrai, il positionne sa sortie R STROBE à l'état vrai. Dans les deux cas, il retourne à l'état de repos.

5.4 Module EMETTEUR DE DONNEES

REGLE 5.5:

Un EMETTEUR DE DONNEES DOIT être apparié avec un RECEPTEUR D'EN-TETE.

La fonction d'un EMETTEUR DE DONNEES est de prendre les données à partir des couches supérieures et d'envoyer les données sur le bus série lorsque le RECEPTEUR D'EN-TETE lui commande de faire ainsi.

5.4.1 Interface avec la couche physique

Comme indiqué dans la figure 5-5, page 138, l'interface d'un EMETTEUR DE DONNEES avec la couche physique inclut les signaux SCLK, XONE et RONE.

5.4.2 Interface avec la couche de liaison de données

Comme indiqué dans la figure 5-5, l'interface d'un EMETTEUR DE DONNEES avec la couche de liaison de données inclut les signaux suivants:

S SELECT	Une entrée à partir du RECEPTEUR D'EN-TETE apparié qui signale quand une donnée doit être envoyée.
DSAE	Une entrée à partir du RECEPTEUR D'EN-TETE apparié qui indique comment la donnée doit être envoyée. Si le signal est à l'état vrai, l'EMETTEUR DE DONNEES valide la logique d'arbitrage du bus série lorsqu'il envoie les données.

5.4.3 Interface avec la couche supérieure

Comme indiqué dans la figure 5-5, l'interface de l'EMETTEUR DE DONNEES avec la couche supérieure inclut les signaux suivants:

RESET*	Lorsque ce signal est activé, l'EMETTEUR DE DONNEES est initialisé comme décrit dans le paragraphe suivant.
--------	---

- 20) Before the next C1 edge, the HEADER RECEIVER makes XONE False and then monitors RONE after the next S1 edge. This is the Jam Detect bit.
- 21) If the Jam Detect bit is a one_bit, or the previously stored MSB of the Frame Status field was a one_bit, the HEADER RECEIVER does not make its S or R STROBE outputs True, but simply returns to idle state.
- 22) (The Jam Detect bit and the previously stored MSB of the Frame Status are both zero_bits.) If S MATCH is True, the HEADER RECEIVER makes its S STROBE output True on the following C1 edge. On the same edge, if R MATCH is True, it makes its R STROBE output True. In either case it returns to idle state.

5.4 DATA SENDER module

RULE 5.5:

A DATA SENDER MUST be paired with a HEADER RECEIVER.

The function of a DATA SENDER is to take data from higher layers and send the data on the serial bus when it is signalled to do so by its HEADER RECEIVER.

5.4.1 Physical Layer Interface

As shown in Figure 5-5, page 139, the Physical Layer interface of a DATA SENDER includes the SCLK, XONE, and RONE signals.

5.4.2 Link Layer Interface

As shown in Figure 5-5, the Link Layer interface of a DATA SENDER includes the following signals:

S SELECT An input from the paired HEADER RECEIVER, which signals when data is to be sent.

DSAE An input from the paired HEADER RECEIVER, which signals how data is to be sent. If this signal is True, the DATA SENDER enables serial bus arbitration logic while sending data.

5.4.3 Higher Layer Interface

As shown in Figure 5-5, the Higher Layer interface of a DATA SENDER includes the following signals:

RESET* When this signal is asserted, the DATA SENDER is initialized as described in Paragraph 5.4.4.

Port de données Un ensemble de lignes d'entrée parallèles et de signaux de commande associés, à travers lesquels les couches supérieures du gestionnaire du bus série fournissent les données à envoyer sur le bus série.

AUTORISATION 5.3:

Le port de données PEUT être implémenté comme des entrées dédiées, ou comme un interface de bus vers un registre de données interne.

OBSERVATION 5.14:

Les trames sur le bus série peuvent inclure 1, 2, 4, 8, 16 ou 32 octets de données.

AUTORISATION 5.4:

Les réalisations de l'EMETTEUR DE DONNEES PEUVENT se limiter à l'envoi de moins de 32 octets.

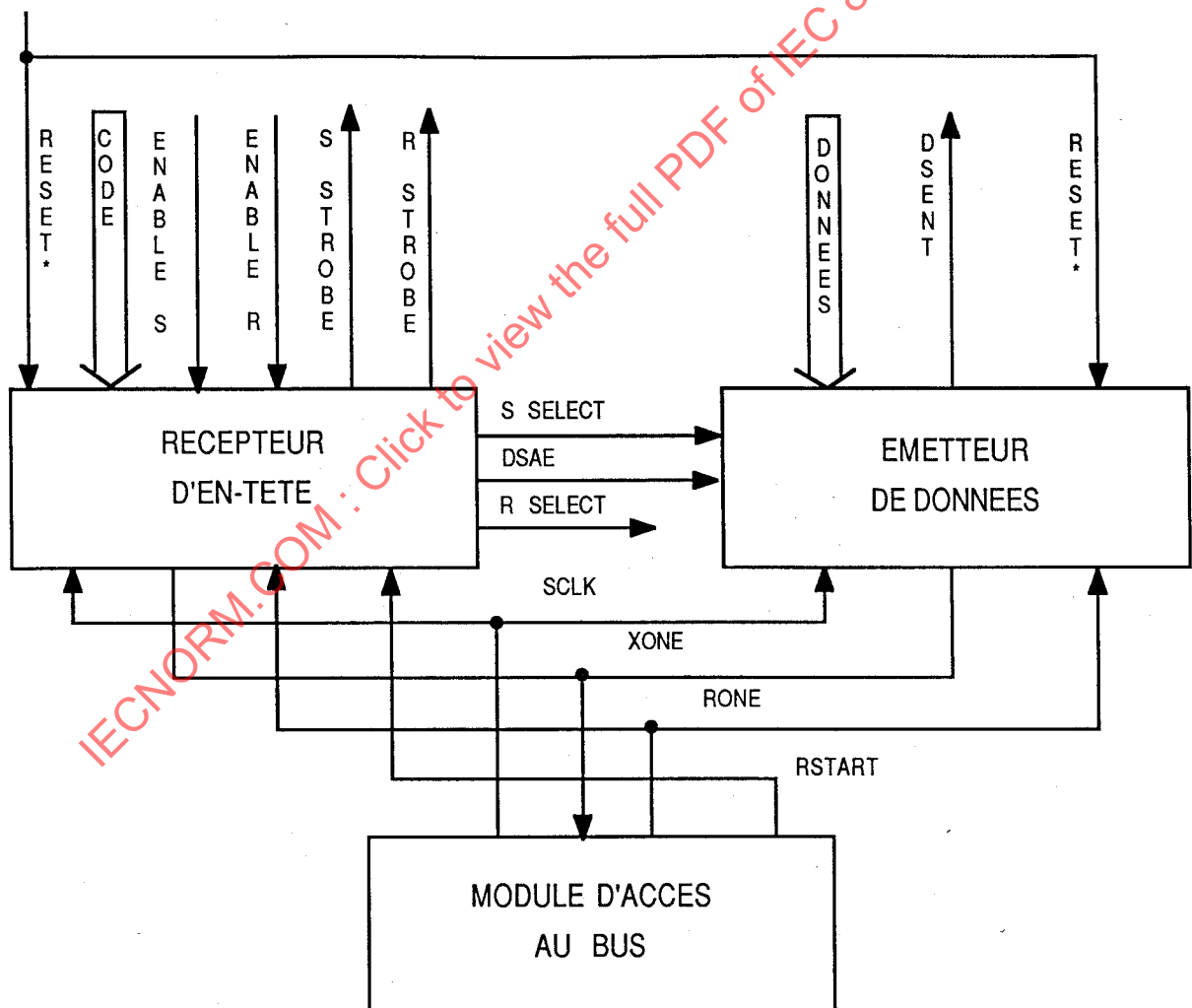


Figure 5-5 - EMETTEUR DE DONNEES avec RECEPTEUR D'EN-TETE

Data Port

A set of parallel input lines and associated control signals, through which higher layers of serial bus management provides data to be sent on the serial bus.

PERMISSION 5.3:

The Data Port MAY be implemented as dedicated inputs, or as a bus interface to an internal Data register.

OBSERVATION 5.14:

Frames on the serial bus can include 1, 2, 4, 8, 16 or 32 bytes of data.

PERMISSION 5.4:

DATA SENDER implementations MAY be limited to sending less than 32 bytes.

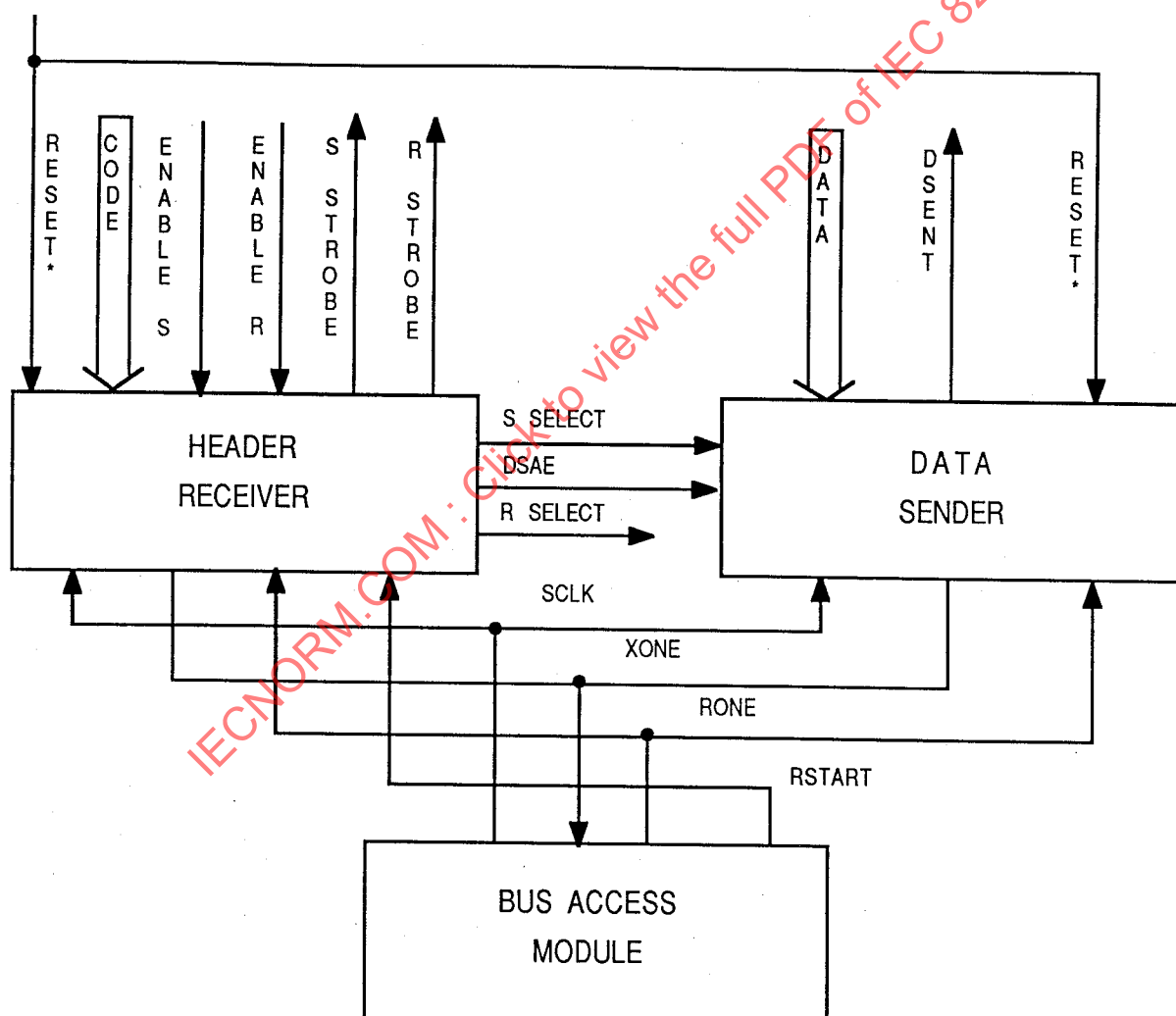


Figure 5-5 - DATA SENDER with HEADER RECEIVER

Un EMETTEUR DE DONNEES signale combien d'octets vont être émis avant d'envoyer les données. Cette norme ne donne pas les détails sur le chargement des données via le port de données, ni comment le nombre d'octets chargés est déterminé par l'EMETTEUR DE DONNEES.

DSENT

Sortie qui signale que l'EMETTEUR DE DONNEES a envoyé les données sur le bus série avec succès.

OBSERVATION 5.15:

Pour les trames qui sélectionnent un seul EMETTEUR DE DONNEES, et les trames sans arbitrage du champ de données (DSAE faux), DSENT actionne de la même manière la sortie S STROBE du RECEPTEUR D'EN-TETE apparié. Cependant, quand une trame sélectionne plusieurs EMETTEURS DE DONNEES pour leur plus grande valeur, tous les RECEPTEURS D'EN-TETE appariés actionneront S STROBE mais seul l'EMETTEUR DE DONNEES gagnant actionnera DSENT.

5.4.4 Initialisation

Chaque fois que RESET* est au niveau bas, un EMETTEUR DE DONNEES s'initialise comme suit:

- 1) Il positionne sa sortie DSENT fausse.
- 2) Il positionne XONE faux et ignore RONE comme entrée.
- 3) Si le port de données est réalisé sous forme de registre chargeable, il met ce registre à zéro.

Quand RESET* va à l'état haut, l'EMETTEUR DE DONNEES entre dans l'état de repos décrit ci-dessous.

5.4.5 Programmation du port de données

Les EMETTEURS DE DONNEES peuvent être utilisés de deux manières: dans la première approche, l'EMETTEUR DE DONNEES est toujours prêt à envoyer la donnée et il envoie "n'importe quelle donnée disponible" chaque fois qu'une trame qui le sélectionne apparaît sur le bus série. Ce type d'application est décrit dans le chapitre 6 comme un groupe "parleur sur demande". Avec ce groupe, les couches supérieures du gestionnaire du bus série présentent la nouvelle donnée sur le port de données "chaque fois que les données envoyées changent". Le "parleur sur demande" sera conçu avec un port de données "double registre" pour assurer que lorsqu'il envoie la donnée dans une trame, il n'envoie pas un mélange d'anciennes et de nouvelles données.

A DATA SENDER signals how many bytes are being sent before it sends data. This standard does not cover the details of how data is loaded via Data Port, nor how the number of bytes loaded is determined by the DATA SENDER.

DSENT

An output which signals that the DATA SENDER has successfully sent data on the serial bus.

OBSERVATION 5.15:

For frames which select a single DATA SENDER, and frames without Data field arbitration (DSAE False), DSENT signals identically to the S STROBE output of the paired HEADER RECEIVER. However, when a frame selects multiple DATA SENDERS for their largest value, all of their paired HEADER RECEIVERS will signal on S STROBE, but only the winning DATA SENDER will signal on DSENT.

5.4.4 Initialization

Whenever RESET* is Low, a DATA SENDER initializes itself as follows:

- 1) It makes its DSENT output False.
- 2) It makes XONE False, and ignores RONE as an input.
- 3) If Data Port is implemented as a loadable register, it clears that register to zero.

When RESET* goes high, the DATA SENDER enters idle state, as described below.

5.4.5 Programming the Data Port

DATA SENDERS can be used in two ways. In one approach, the DATA SENDER is always ready to send data, and sends "whatever data is available" whenever a frame which selects it appears on the serial bus. This type of application is described in Chapter 6 as an "On-Demand Talker" group. With this group, higher layers of serial bus management present new data at the Data Port "whenever the data to be sent changes". The On-Demand Talker design should "double buffer" the Data Port to ensure that when it sends data in a frame, it does not send a mixture of old and new data.

Dans d'autres applications, chaque fois qu'une nouvelle donnée est présentée au port de données, l'EMETTEUR DE DONNEES l'envoie (avec succès) *une fois et seulement une fois*. Ce type d'application est décrit dans le chapitre 6 comme un groupe de modules "parleur transactionnel". Avec un tel groupe, les couches supérieures du gestionnaire du bus série présentent la nouvelle donnée, chaque fois qu'elle est disponible et que la donnée chargée préalablement a été envoyée avec succès. Si une trame sélectionnant un tel EMETTEUR DE DONNEES apparaît sur le bus série, alors qu'une nouvelle donnée n'a pas été fournie, l'EMETTEUR DE DONNEES annule la trame.

Pour permettre à un parleur transactionnel d'opérer de cette manière, un EMETTEUR DE DONNEES commande la sortie DSENT à l'état vrai seulement quand tous les points suivants sont vrais:

- 1) S'il n'a pas perdu l'arbitrage du champ de donnée (cela s'applique seulement si DSAE est vrai).
- 2) Le champ d'état de la trame contient 011.
- 3) La sous-trame de détection de désynchronisation de trame contient un bit_zéro.

Si un parleur transactionnel envoie des données, alors que DSENT n'est pas positionné parce qu'une ou plusieurs des conditions ci-dessus ne sont pas remplies, le groupe retient ses données pour une retransmission dans une autre trame.

La description suivante d'un EMETTEUR DE DONNEES convient aux deux groupes "parleur transactionnel" et "sur demande". Dans le premier cas, l'entrée ENABLE S du RECEPTEUR D'EN-TETE apparié est en permanence à l'état vrai et la sortie DSENT de l'EMETTEUR DE DONNEES n'est pas utilisée. Dans le second cas, les couches supérieures du gestionnaire du bus série mettent ENABLE S à l'état vrai après que la donnée a été chargée et ENABLE S redevient faux quand l'EMETTEUR DE DONNEES génère une impulsion sur sa sortie DSENT.

5.4.6 Fonctionnement

REGLE 5.6:

Après avoir effectué la procédure d'initialisation ci-dessus, un EMETTEUR DE DONNEES DOIT opérer comme défini par le diagramme d'état de la figure 5-6, page 146, et la description narrative suivante.

En partant de l'état de repos, un EMETTEUR DE DONNEES opère ainsi:

- 1) A l'état de repos, l'EMETTEUR DE DONNEES échantillonne son entrée S SELECT sur chaque transition S1 de SCLK et surveille ensuite RONE. Quand il échantillonne S SELECT à l'état vrai et surveille RONE vrai, il passe à l'état 2.

In other applications, each time new data is presented at the Data Port, the DATA SENDER sends it (successfully) *once and only once*. This type of application is described in Chapter 6 as a "Transaction Talker" module group. With such a group, higher layers of serial bus management present new data whenever it is available *and* previously loaded data has been successfully sent. If a frame which selects such a DATA SENDER appears on the serial bus, and new data has not been provided, the DATA SENDER cancels the frame.

To allow a Transaction Talker to operate in this way, a DATA SENDER makes its DSENT output True only when all of the following are true:

- 1) If it did not lose Data Field arbitration (this applies only if DSAE is True).
- 2) The Frame Status field contains 011.
- 3) The Jam Detect subframe contains a zero_bit.

If a Transaction Talker sends data, but DSENT is not asserted because one or more of the above conditions is not met, the group retains its data for retransmission in a future frame.

The following description of a DATA SENDER allows for both On-Demand and Transaction Talker groups. In the first case, the ENABLE S input of the paired HEADER RECEIVER is made permanently True and the DSENT output of the DATA SENDER is not used. In the second case, higher layers of serial bus management make ENABLE S True after data has been loaded, and ENABLE S goes False again when the DATA SENDER pulses its DSENT output.

5.4.6 Operation

RULE 5.6:

After completing the above initialization procedure, a DATA SENDER MUST operate as defined by the state diagram in Figure 5-6, page 147, and the following narrative description.

Starting from idle state, a DATA SENDER proceeds as follows:

- 1) In idle state the DATA SENDER samples its S SELECT input on every S1 transition of SCLK and monitors RONE thereafter. When it samples S SELECT True and monitors RONE True, it proceeds to step 2.

OBSERVATION 5.16:

Cette description suppose que l'EMETTEUR DE DONNEES "connaît" le nombre d'octets fournis par les couches supérieures du gestionnaire du bus série la dernière fois que de nouvelles données ont été présentées au port de données. De plus, il suppose que l'EMETTEUR DE DONNEES les a converties en un code de type de trame de 3 bits comme suit:

<i>Nombre d'octets</i>	<i>Type de trame (3 bits)</i>
1	001
2	010
3-4	011
5-8	100
9-16	101
17-31	110

- 2) Avant le front C1 suivant de SCLK, l'EMETTEUR DE DONNEES place le bit le plus significatif de ses 3 bits de type de trame sur XONE.
- 3) Après le front S1 suivant de SCLK, il surveille et stocke l'état de RONE. S'il avait commandé XONE à l'état vrai ou s'il détecte RONE à l'état faux, l'EMETTEUR DE DONNEES agit pour placer le deuxième et troisième bits de ce code sur XONE, avant les fronts C1 suivants.

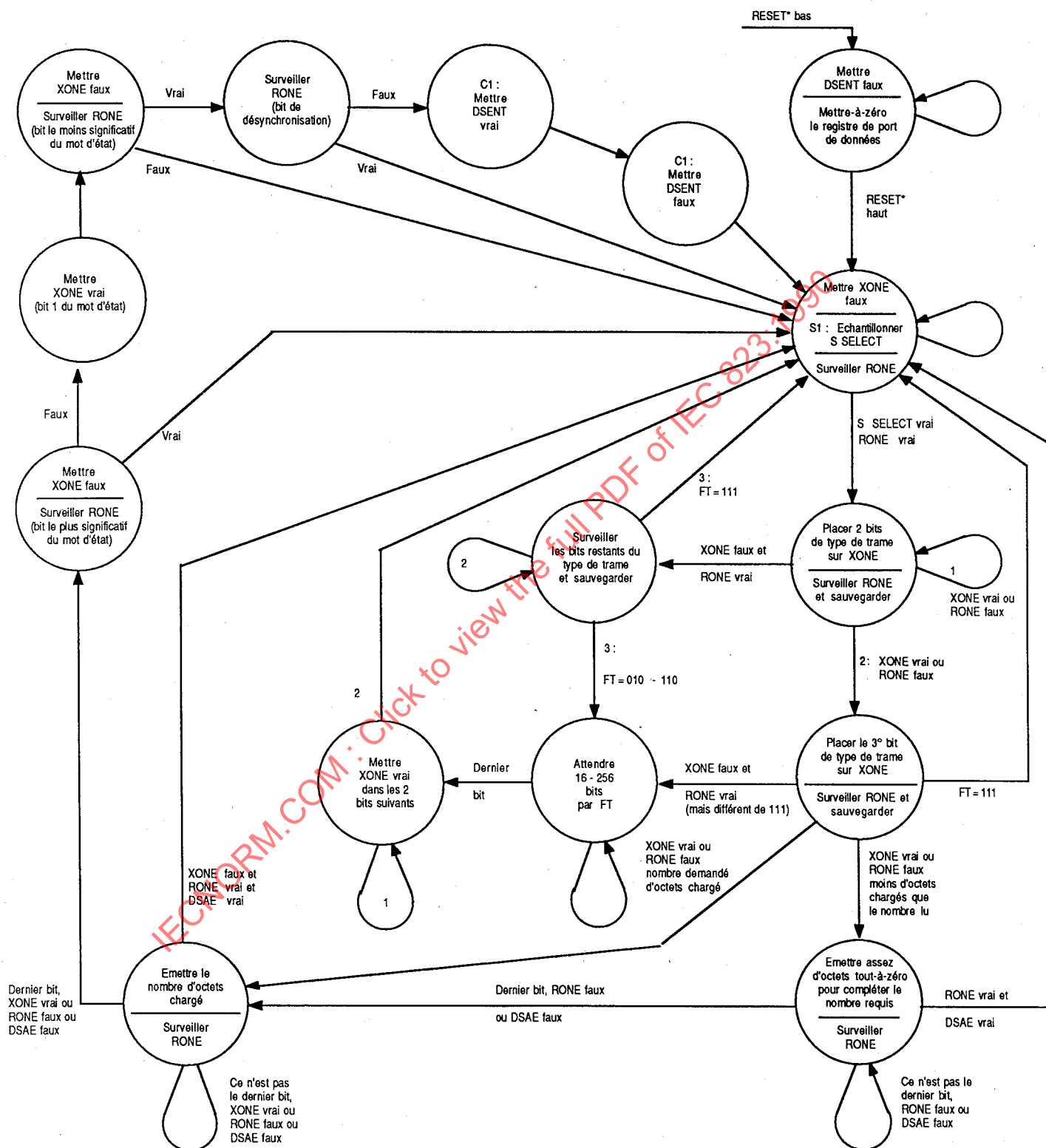
OBSERVATION 5.16:

This description assumes that the DATA SENDER "knows" the number of bytes provided by higher layers of serial bus management the last time new data was presented at Data Port. It further assumes that the DATA SENDER has converted this to a 3-bit Frame Type code as follows:

<i>Number of bytes</i>	<i>3-bit Frame Type</i>
1	001
2	010
3-4	011
5-8	100
9-16	101
17-32	110

- 2) Before the next C1 edge of SCLK, the DATA SENDER places the most significant bit of its 3-bit Frame Type on XONE.
- 3) After the next S1 edge on SCLK, it monitors and stores the state of RONE. If it made XONE True or if it monitors RONE False, the DATA SENDER proceeds to place the second and third bits of its code on XONE, before subsequent C1 edges.

IECNORM.COM : Click to view the full PDF of IEC 823:1990



FT = Type de trame

Figure 5-6 - Diagramme d'état de l'EMETTEUR DE DONNEES



Figure 5-6 - DATA SENDER state diagram

- 4) Si sur l'un de ces trois bits, il voit RONE vrai après avoir mis XONE à l'état faux, l'EMETTEUR DE DONNEES met XONE à l'état faux pour chacun des bits restant de ses 3 bits de code, mais il continue à surveiller et stocker chacun des bits de type de trame restant après les fronts S1.
- 5) Après le troisième front S1, si les trois bits stockés représentent 111, la trame a été annulée par un RECEPTEUR D'EN-TETE. Dans ce cas, l'EMETTEUR DE DONNEES retourne simplement à l'état de repos.
- 6) Si les trois bits stockés ne sont pas 111, mais sont différents du code que l'EMETTEUR DE DONNEES s'apprête à envoyer, il y a un "conflit de taille émetteur/émetteur". Dans ce cas, l'EMETTEUR DE DONNEES utilise le code stocké pour déterminer le nombre d'octets de données dans la trame, laisse passer le nombre correspondant de bits sur SCLK et place alors 110 sur XONE avant les trois prochains fronts C1 (c'est-à-dire dans le champ d'état de la trame). (Il ne surveille pas RONE ni n'arbitre dans le champ d'état.) Il retourne alors à l'état de repos.
- 7) Si les trois bits stockés coïncident avec le type de trame de l'EMETTEUR DE DONNEES, il commence à envoyer les données sur XONE. Si le nombre d'octets chargés par les couches supérieures du gestionnaire du bus série n'est pas une puissance de deux, l'EMETTEUR DE DONNEES envoie d'abord le nombre d'octets *tout-à-zéro* nécessaires pour compléter le nombre d'octets à envoyer (c'est-à-dire que la donnée chargée est émise "cadrée à droite").

Après les octets tout-à-zéro, ou immédiatement si les couches supérieures ont fourni le nombre d'octets correct, il place les bits du port de donnée sur XONE, commençant avec le bit le plus significatif de l'octet le plus significatif (le plus à gauche) et terminant avec le bit le moins significatif de l'octet le moins significatif (le plus à droite).
- 8) Lorsque l'EMETTEUR DE DONNEES place chaque bit de donnée sur XONE, il surveille RONE après le front S1 suivant. S'il voit RONE vrai après avoir commandé XONE faux, et si son entrée DSAE est vraie, alors il retourne immédiatement à l'état de repos. S'il voit RONE faux, ou s'il a commandé XONE vrai, ou si l'entrée DSAE est fautive, il continue d'envoyer les bits de données.
- 9) Après que l'EMETTEUR DE DONNEES a identifié le dernier bit de données, il commande XONE faux avant le prochain front C1 de SCLK et surveille alors RONE après le prochain front S1. C'est le bit 2 du champ d'état de la trame. S'il voit RONE vrai il retourne à l'état de repos.
- 10) (Le bit 2 de l'état de la trame est un bit_zéro.) L'EMETTEUR DE DONNEES commande alors XONE vrai avant le prochain front C1 pour montrer qu'il est sélectionné et qu'il a survécu à l'arbitrage du champ de données. Avant le front C1 suivant, il commande XONE faux, et alors surveille et stocke RONE après les deux fronts S1 suivants de SCLK. Ce sont le bit 0 du champ d'état de la trame et le bit détection de désynchronisation.

- 4) If, on any of these three bits, it monitors RONE True after it made XONE False, the DATA SENDER makes XONE False for any remaining bits of its 3-bit code, but it continues to monitor and store any remaining Frame Type bits after S1 edges.
- 5) After the third S1 edge, if the three stored bits are 111, the frame has been cancelled by a HEADER RECEIVER. In this case, the DATA SENDER simply returns to idle state.
- 6) If the three stored bits are not 111, but are not the same as the code the DATA SENDER set out to send, there is a "Sender/Sender Size Conflict". In this case, the DATA SENDER uses the stored code to determine the number of data bytes in the frame, counts off the corresponding number of bits on SCLK and then places 110 on XONE before the next three C1 edges (i.e. in the Frame Status field). (It does not monitor RONE nor arbitrate in the Status field.) It then returns to idle state.
- 7) If the three stored bits match the DATA SENDER's Frame Type, it begins to send data on XONE. If the number of bytes loaded by higher layers of serial bus management is not a power of two, the DATA SENDER first sends the number of *all-zero* bytes necessary to fill out the number of bytes to be sent (i.e. the loaded data is sent "right justified").

After the all-zero bytes, or immediately if higher layers provided the "right" number of bytes, it places bits from the Data Port on XONE, starting with the most significant bit of the most significant (leftmost) byte and ending with the least significant bit of the least significant (rightmost) byte.

- 8) As the DATA SENDER places each bit of data on XONE, it monitors RONE after the following S1 edge. If it monitors RONE True after it made XONE False, *and* if its DSAE input is True, then it immediately returns to idle state. If it monitors RONE False, or if it made XONE True, or if the DSAE input is False, it continues sending data bits.
- 9) After the DATA SENDER has monitored the last data bit, it makes XONE False before the next C1 edge of SCLK and then monitors RONE after the next S1 edge. This is bit 2 of the Frame Status field. If it monitors RONE True it returns to idle state.
- 10) (Bit 2 of the Frame Status is a zero_bit.) The DATA SENDER then makes XONE True before the next C1 edge, to show that it is selected and that it has survived any Data field arbitration. Before the following C1 edge it makes XONE False, and then monitors and stores RONE after the next two S1 edges of SCLK. These are bit 0 of the Frame Status field and the Jam Detect bit.

- 11) Si le bit 0 de l'état de la trame est un bit_zéro, ou le bit de détection de désynchronisation est un bit_un, l'EMETTEUR DE DONNEES retourne à l'état de repos.
- 12) Si ces 2 bits sont 10, l'EMETTEUR DE DONNEES commande sa sortie DSENT vraie au front C1 suivant pour signaler à la logique de la couche supérieure qu'il a envoyé la donnée avec succès. Sur le front C1 suivant, il commande de nouveau DSENT à l'état faux et retourne à l'état de repos.

5.5 Module RECEPTEUR DE DONNEES

REGLE 5.7:

Un RECEPTEUR DE DONNEES DOIT être apparié avec un RECEPTEUR D'EN-TETE.

La fonction d'un RECEPTEUR DE DONNEES est de saisir la donnée envoyée par l'EMETTEUR DE DONNEES sur le bus série quand son RECEPTEUR D'EN-TETE le lui demande, et de présenter la donnée aux couches supérieures du gestionnaire du bus série.

5.5.1 Interface avec la couche physique

Comme le montre la figure 5-7, page 152, l'interface avec la couche physique d'un RECEPTEUR DE DONNEES inclut les signaux SCLK, XONE et RONE.

5.5.2 Interface avec la couche de liaison de données

Comme le montre la figure 5-7, l'interface avec la couche de liaison d'un RECEPTEUR DE DONNEES inclut le signal suivant:

R SELECT	Entrée venant du RECEPTEUR D'EN-TETE apparié qui sélectionne le RECEPTEUR DE DONNEES pour saisir la donnée sur le bus série.
----------	--

5.5.3 Interface avec les couches de niveau supérieur

Comme le montre la figure 5-7, l'interface de la couche supérieure d'un RECEPTEUR DE DONNEES inclut les signaux suivants:

OBSERVATION 5.17:

La sortie R STROBE du RECEPTEUR D'EN-TETE apparié signale aux couches supérieures que la donnée du bus série est disponible sur le port de données.

RESET*	Quand ce signal est activé, le RECEPTEUR DE DONNEES est initialisé comme décrit dans le paragraphe suivant.
--------	---

Port de donnée	Ensemble de lignes de sortie parallèles et de signaux de contrôle associés à travers lesquelles la donnée du bus série est présentée aux couches supérieures du gestionnaire du bus série.
----------------	--

- 11) If bit 0 of the Frame Status is a zero_bit, or the Jam Detect bit is a one_bit, the DATA SENDER simply returns to idle state.
- 12) If these 2 bits are 10, the DATA SENDER makes its DSENT output True at the next C1 edge, to signal to higher-layer logic that it has sent data successfully. On the following C1 edge it makes DSENT False again and returns to idle state.

5.5 DATA RECEIVER module

RULE 5.7:

A DATA RECEIVER MUST be paired with a HEADER RECEIVER.

The function of a DATA RECEIVER is to capture data sent by DATA SENDERS on the serial bus when it is signalled to do so by its HEADER RECEIVER, and to present the data to higher layers of serial bus management.

5.5.1 Physical Layer interface

As shown in Figure 5-7, page 153, the Physical Layer interface of a DATA RECEIVER includes the SCLK, XONE and RONE signals.

5.5.2 Link Layer interface

As shown in Figure 5-7, the Link Layer interface of a DATA RECEIVER includes the following signal:

R SELECT	An input from the paired HEADER RECEIVER, which selects the DATA RECEIVER to capture data from the serial bus.
----------	--

5.5.3 Higher Layer interface

As shown in Figure 5-7, the Higher Layer interface of a DATA RECEIVER includes the following signals:

OBSERVATION 5.17:

The R STROBE output of the paired HEADER RECEIVER signals higher layers when data from the serial bus is available on Data Port.

RESET*	When this signal is asserted, the DATA RECEIVER is initialized as described in the following Paragraph 5.5.4.
--------	---

Data Port	A set of parallel output lines and associated control signals, through which data from the serial bus is presented to higher layers of serial bus management.
-----------	---

OBSERVATION 5.18:

Les trames sur le bus série peuvent inclure 1, 2, 4, 8, 16 ou 32 octets de données.

AUTORISATION 5.5:

Les réalisations de RECEPTEURS DE DONNEES PEUVENT se limiter à la réception de moins de 32 octets.

Si une trame sélectionne un RECEPTEUR DE DONNEES et si elle contient plus de données que le RECEPTEUR DE DONNEES ne peut en accepter, le module signale le problème sur le bus série dans le champ d'état de la trame. Cette norme ne fournit pas les détails sur la manière dont les couches de haut niveau lisent la donnée reçue du port de données.

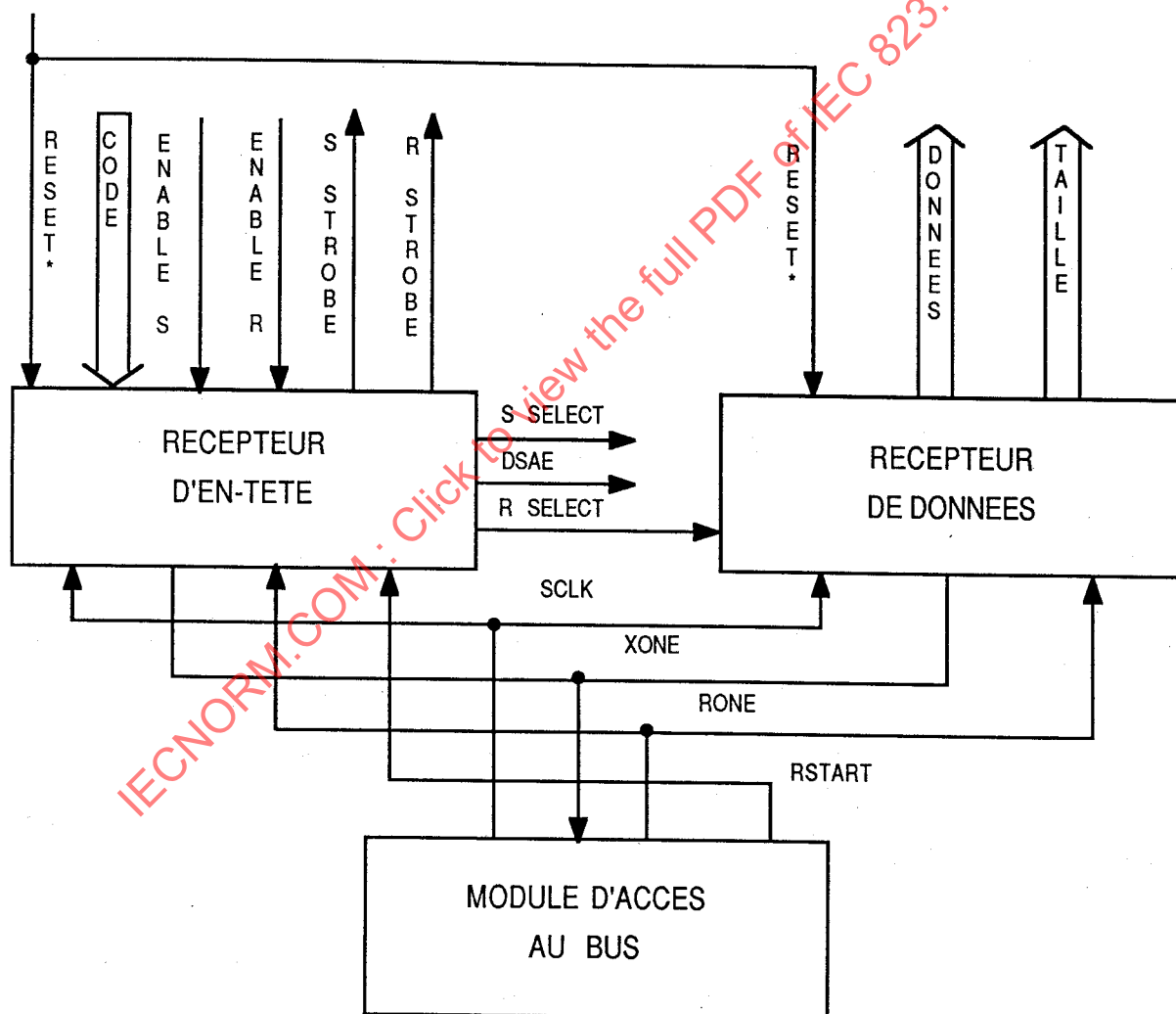


Figure 5-7 - RECEPTEUR DE DONNEES avec RECEPTEUR D'EN-TETE

OBSERVATION 5.18:

Frames on the serial bus can include 1, 2, 4, 8, 16 or 32 bytes of data.

PERMISSION 5.5:

DATA RECEIVER implementations MAY be limited to receiving less than 32 bytes.

If a frame selects a DATA RECEIVER and it contains more data than the DATA RECEIVER can accept, the module signals the problem on the serial bus in the Frame Status field. This standard does not cover the details of how higher layers read received data from Data Port.

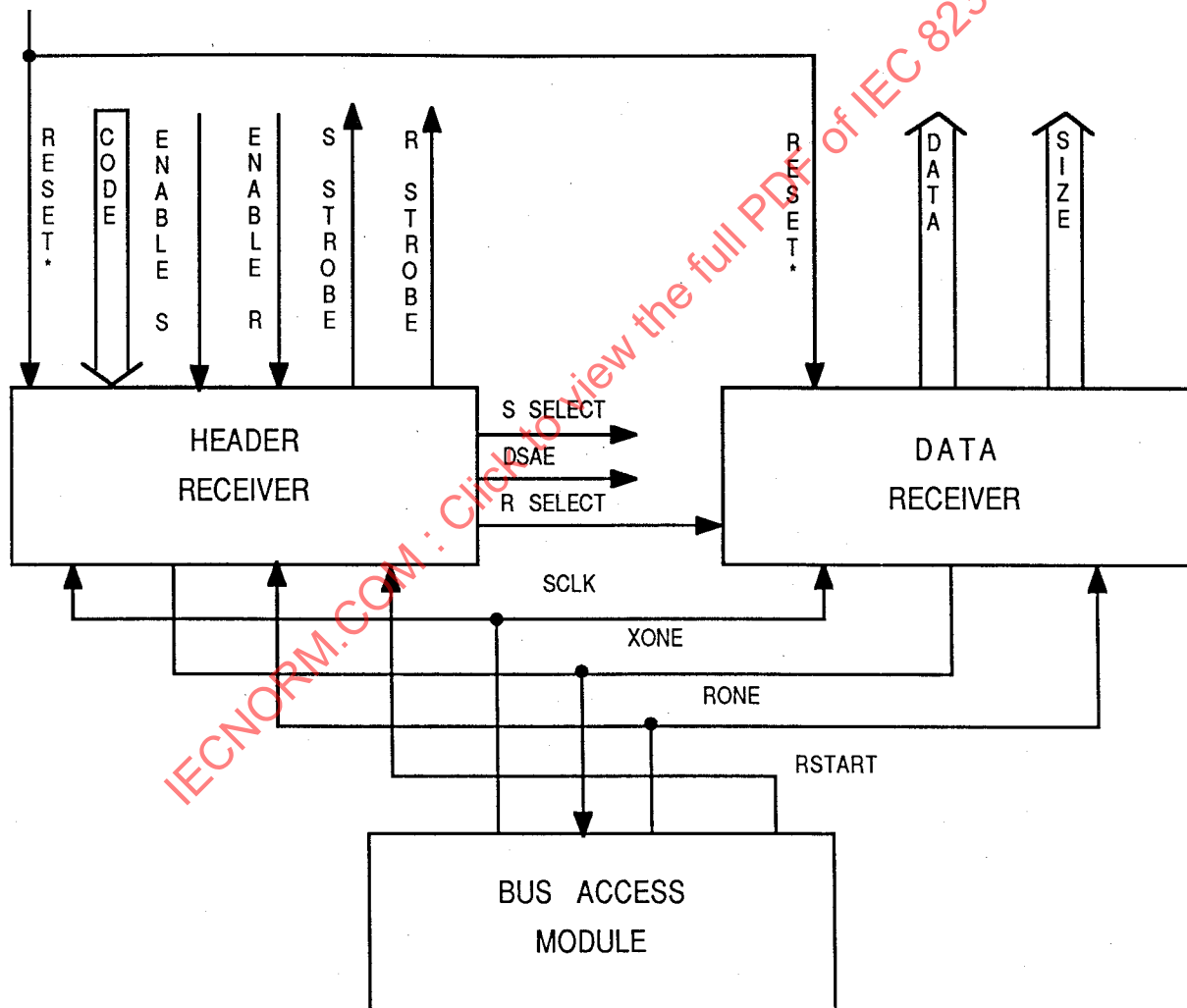


Figure 5-7 - DATA RECEIVER with HEADER RECEIVER

Port de taille de données Trois sorties parallèles et les signaux associés à travers lesquelles le nombre d'octets reçu est présenté aux couches supérieures du gestionnaire du bus série. Ces sorties sont codées comme décrit ailleurs dans cette norme pour la sous-trame de type de trame.

PERMISSION 5.6:

Le port de données et le port de taille de données PEUVENT être réalisés par signaux dédiés ou par un interface de bus commun.

5.5.4 Initialisation

Chaque fois que RESET* est au niveau bas, un RECEPTEUR DE DONNEES s'initialise comme suit:

- 1) Il commande XONE faux et ignore RONE comme entrée.
- 2) Si le port de données et/ou le port de taille de données est réalisé comme un registre chargeable, il met à zéro le ou les registres.

Quand RESET* passe au niveau haut, le RECEPTEUR DE DONNEES entre dans l'état de repos décrit ci-dessous.

5.5.5 Lecture des données du port de données

Dans un système, les RECEPTEURS DE DONNEES peuvent être utilisés de deux façons. Dans la première approche, le RECEPTEUR DE DONNEES est toujours prêt à recevoir la donnée, et met la donnée disponible sur le port de donnée chaque fois qu'il est sélectionné par une trame sur le bus série. Ce type d'application est décrit dans le chapitre 6 comme un groupe "écouteur sur demande". Avec un tel groupe, les couches supérieures du gestionnaire du bus série peuvent utiliser la donnée du port de donnée "chaque fois que c'est nécessaire". Dans ce cas, si les couches supérieures utilisent plusieurs "cycle lecture" pour lire la donnée et qu'une nouvelle donnée arrive sur le bus série alors tandis qu'il est en train de lire, les couches supérieures peuvent recevoir un mélange d'anciennes et de nouvelles données. Les réalisations écouteur sur demande ont à traiter ce problème. Une manière de faire est de le signaler aux couches supérieures du gestionnaire du bus série si le problème apparaît.

Dans d'autres applications, chaque fois qu'une nouvelle donnée arrive (correctement) sur le bus série, les couches supérieures du gestionnaire du bus série la lisent sur le port de données *une fois et seulement une fois*. Ce type d'application est décrit dans le chapitre 6 comme un "écouteur transactionnel". Dans ce schéma, quand la donnée arrive du bus série, les couches supérieures la lisent à partir du RECEPTEUR DE DONNEES dès que possible. Si une autre trame de transfert de données sélectionne le RECEPTEUR DE DONNEES avant que la donnée précédente ait été lue, le module annule la trame.

Data Size Port

Three parallel outputs and associated control signals, through which the number of received bytes is presented to higher layers of serial bus management. These outputs are encoded as described elsewhere in this standard for the Frame Type subframe.

PERMISSION 5.6:

Data Port and Data Size Port MAY be implemented by dedicated signals, or by a common bus interface.

5.5.4 Initialization

Whenever RESET* is Low, a DATA RECEIVER initializes itself as follows:

- 1) It makes XONE False and ignores RONE as an input.
- 2) If the Data Port and/or Data Size Port is implemented as a loadable register, it clears the register(s) to zero.

When RESET* goes high, the DATA RECEIVER enters idle state as described below.

5.5.5 Reading data from the Data Port

In a system, DATA RECEIVERS can be used in two ways. In one approach the DATA RECEIVER is always ready to receive data, and changes the data available on Data Port whenever it is selected by a frame on the serial bus. This type of application is described in Chapter 6 as an "On-Demand Listener" group. With such a group, higher layers of serial bus management can use the data from the Data Port "whenever it is needed". In this case, if higher layers use multiple "read cycles" to read out the data, and new data arrives on the serial bus while it is doing so, the higher layers may read out a mixture of old and new data. On-Demand Listener implementations have to deal with this problem. One way to handle it is to signal higher layers of serial bus management if the problem arises.

In other applications, each time new data arrives (successfully) on the serial bus, higher layers of serial bus management read it from Data Port *once and only once*. This type of application is described in Chapter 6 as a "Transaction Listener". In this scheme, when data arrives from the serial bus, higher layers read it out of the DATA RECEIVER as soon as possible. If another Data Transfer frame selects the DATA RECEIVER before the previous data has been read, the module cancels the frame.

Le RECEPTEUR DE DONNEES décrit ici peut être utilisé dans les deux types de groupes. Dans l'écouteur sur demande, l'entrée ENABLE R du RECEPTEUR D'EN-TETE apparié est en permanence positionné à l'état vrai et sa sortie R STROBE n'est pas utilisée. Dans l'écouteur transactionnel, ENABLE R devient faux quand le RECEPTEUR D'EN-TETE génère l'impulsion R STROBE et les couches supérieures du gestionnaire du bus série commandent ENABLE R de nouveau à l'état vrai après que la donnée a été lue.

5.5.6 Fonctionnement

REGLE 5.8:

Après avoir effectué la procédure d'initialisation ci-dessus, un RECEPTEUR DE DONNEES DOIT opérer comme défini par le diagramme d'état de la figure 5-8, page 158, et par la description narrative suivante.

En partant de l'état de repos, le RECEPTEUR DE DONNEES opère ainsi:

- 1) A l'état de repos, le RECEPTEUR DE DONNEES échantillonne son entrée R SELECT à chaque transition S1 de SCLK et surveille RONE après la transition jusqu'à la détection de R SELECT à l'état vrai et de RONE à l'état vrai.
- 2) Après chacun des trois fronts S1 suivants de SCLK, le RECEPTEUR DE DONNEES surveille RONE et mémorise son état. Ces états constituent les bits de type de trame. Le RECEPTEUR DE DONNEES interprète le code de type de trame comme suit:

Type de trame	Interprétation
000	Aucun EMETTEUR DE DONNEES n'est sélectionné
001	1 octet de données dans la trame
010	2 octets de données dans la trame
011	4 octets de données dans la trame
100	8 octets de données dans la trame
101	16 octets de données dans la trame
110	32 octets de données dans la trame
111	Trame annulée

- 3) Si le type de trame est 111, c'est une trame annulée et le RECEPTEUR DE DONNEES retourne simplement à l'état de repos.
- 4) Si le type de trame est 000, aucun EMETTEUR DE DONNEES n'est sélectionné dans le système pour envoyer des données. Dans ce cas, le RECEPTEUR DE DONNEES place 101 sur XONE avant chacun des trois fronts C1 suivants de SCLK pour indiquer l'erreur. Avant le front C1 suivant, il positionne XONE à l'état faux et retourne à l'état repos
- 5) Si le type de trame indique plus de données que le RECEPTEUR DE DONNEES ne peut en manipuler, il attend la fin du nombre d'octets indiqué, et place alors 101 sur XONE avant chacun des trois fronts C1 suivants de SCLK. Avant le front C1 suivant, il positionne XONE à l'état faux et retourne à l'état de repos.

The DATA RECEIVER described here can be used in both types of groups. In the On-Demand Listener, the ENABLE R input of the paired HEADER RECEIVER is permanently set to True and its R STROBE output is not used. In the Transaction Listener, ENABLE R False goes when the HEADER RECEIVER pulses R STROBE and higher layers of serial bus management make ENABLE R True again after the data has been read out.

5.5.6 Operation

RULE 5.8:

After completing the above initialization procedure, a DATA RECEIVER MUST operate as defined by the state diagram in Figure 5-8, page 159, and the following narrative description.

Starting from idle state, a DATA RECEIVER proceeds as follows:

- 1) In idle state the DATA RECEIVER samples its R SELECT input on every S1 transition of SCLK, and monitors RONE after the transition, until it samples R SELECT True and monitors RONE True.
- 2) After each of the next three S1 edges of SCLK, the DATA RECEIVER monitors RONE and stores its state. These are the Frame Type bits. The DATA RECEIVER interprets the Frame Type code as follows:

Frame Type	Interpretation
000	No DATA SENDER selected
001	1 byte of data in the frame
010	2 bytes of data in the frame
011	4 bytes of data in the frame
100	8 bytes of data in the frame
101	16 bytes of data in the frame
110	32 bytes of data in the frame
111	Cancelled frame

- 3) If the Frame Type is 111, this is a Cancelled frame and the DATA RECEIVER simply returns to idle state.
- 4) If the Frame Type code is 000, no DATA SENDER in the system is selected to send data. In this case, the DATA RECEIVER places 101 on XONE before each of the next three C1 edges of SCLK to signal the error. Before the next C1 edge it makes XONE False and returns to idle state.
- 5) If the Frame Type indicates more data than the DATA RECEIVER is designed to handle, it waits out the indicated number of bytes, and then places 101 on XONE before each of the next three C1 edges of SCLK. Before the next C1 edge it makes XONE False and returns to idle state.

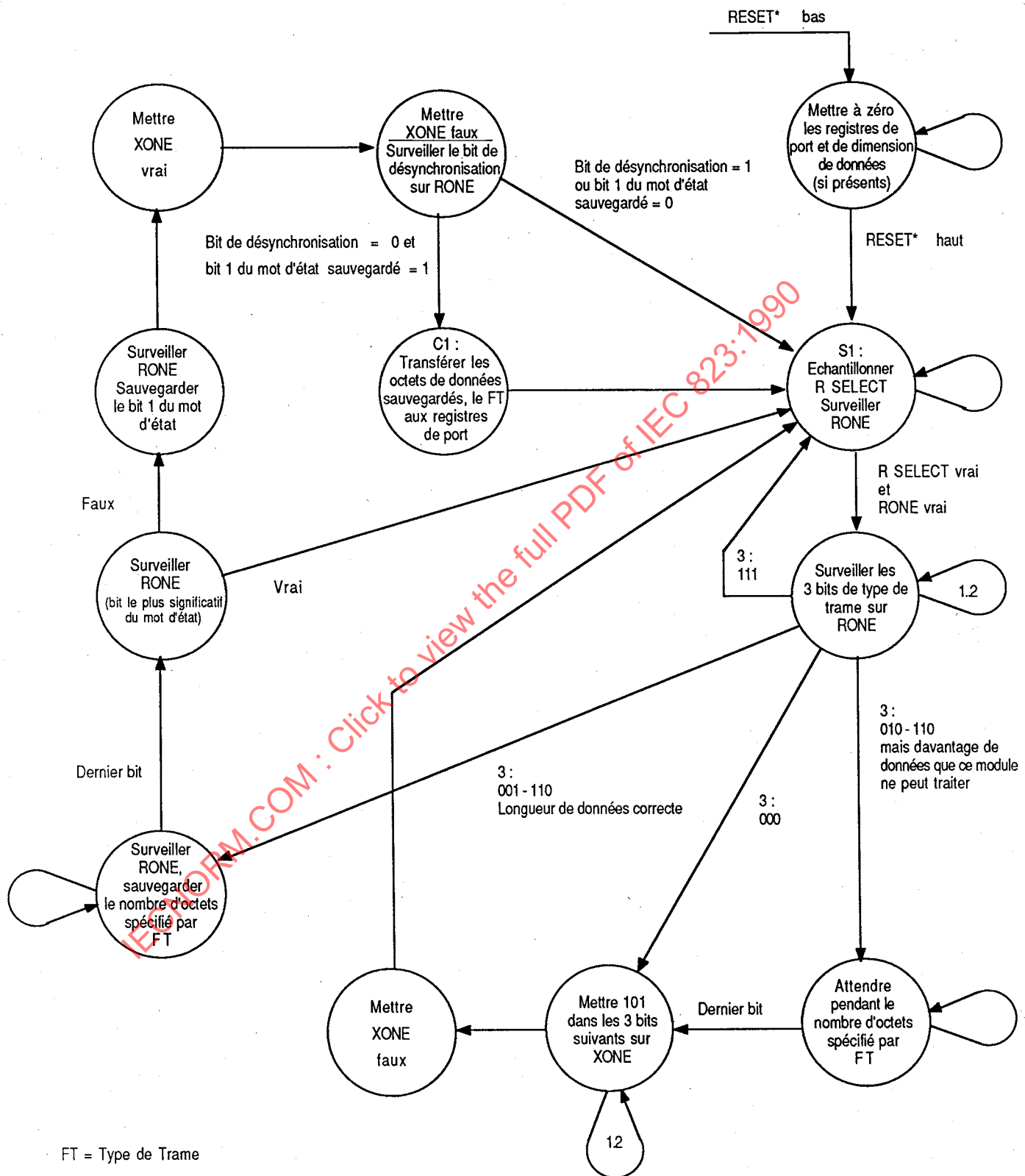


Figure 5-8 - Diagramme d'état du RECEPTEUR DE DONNEES

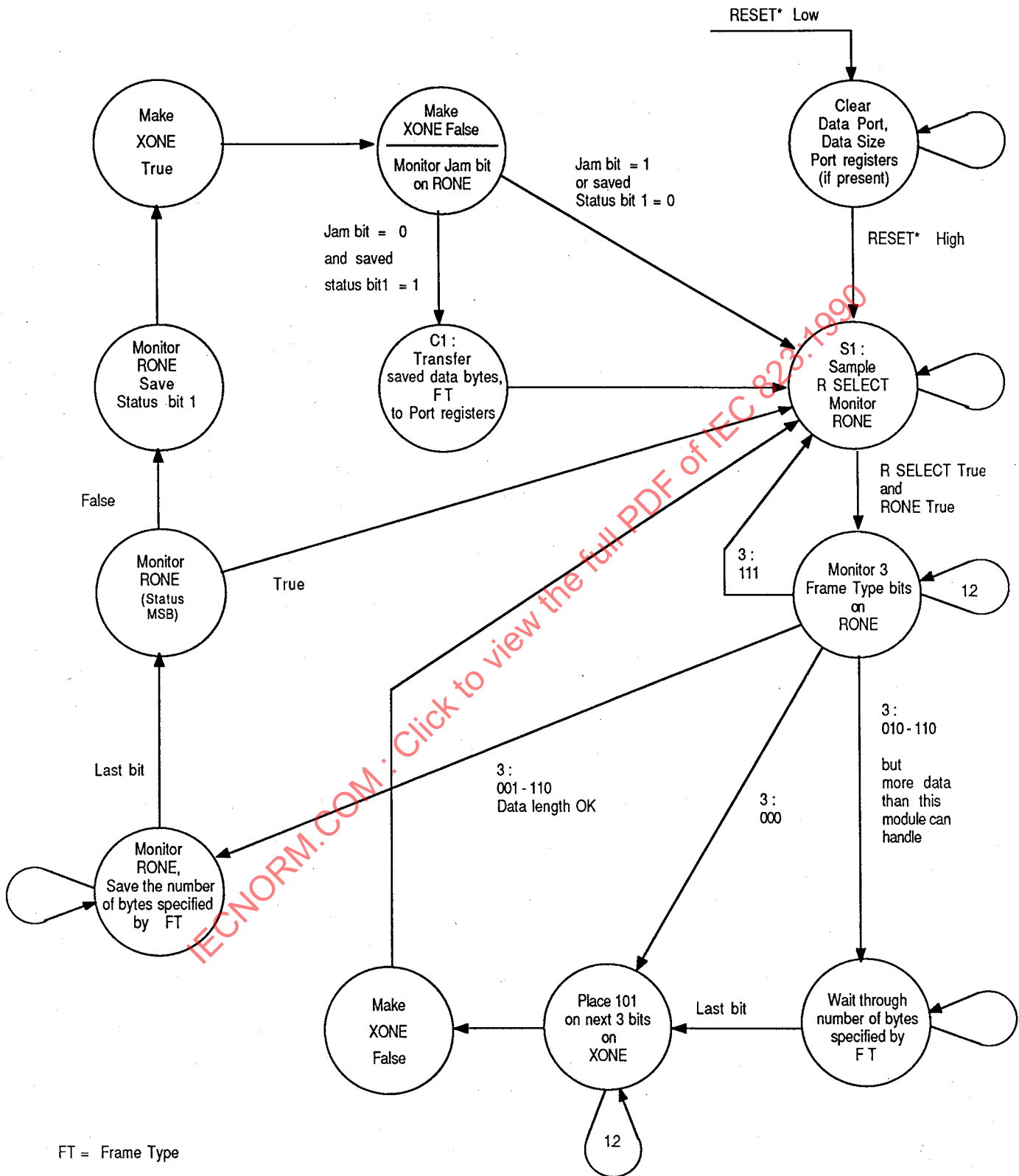


Figure 5-8 - DATA RECEIVER state diagram

- 6) Si le code de type de trame indique une longueur que le RECEPTEUR DE DONNEES peut manipuler, celui-ci surveille et mémorise le nombre d'octets indiqué à partir de RONE, après chaque front S1 suivant de SCLK. Il mémorise la donnée dans le registre de maintien de la donnée, en commençant par le bit le plus significatif de l'octet le plus significatif (le plus à gauche) et en terminant par le bit le moins significatif de l'octet le moins significatif (le plus à droite).

AUTORISATION 5.7:

Dans une réalisation d'un écouteur transactionnel, qui est conçu pour fournir les données aux couches supérieures "une fois et seulement une fois", comme il est décrit aux paragraphes 5.5.5 et 6.1.6, les concepteurs PEUVENT omettre le registre de maintien de donnée et PEUVENT mémoriser la donnée directement dans le registre de données qui est disponible à travers le port de données.

OBSERVATION 5.19:

Dans un écouteur transactionnel, si des problèmes ultérieurs apparaissent dans la sous-trame d'état, la sortie R STROBE du RECEPTEUR D'EN-TETE apparié est simplement non activée et ainsi les couches de niveau supérieur ne sont pas averties que des données nouvelles sont arrivées.

- 7) Après avoir mémorisé le dernier bit de données, le RECEPTEUR DE DONNEES surveille RONE après le front S1 suivant de SCLK. C'est le bit 2 du champ d'état de la trame. S'il détecte RONE à l'état vrai, alors il y a des problèmes avec la trame et le RECEPTEUR DE DONNEES retourne simplement à l'état repos.
- 8) (Le bit 2 du mot d'état de la trame est un bit_zéro.) Après le front S1 suivant de SCLK, le RECEPTEUR DE DONNEES surveille à nouveau RONE (bit 1 du champ d'état de trame) et sauvegarde son état.
- 9) Avant le front C1 suivant de SCLK, le RECEPTEUR DE DONNEES positionne XONE à l'état vrai pour indiquer qu'il est sélectionné. C'est le bit 0 du champ d'état de trame.
- 10) Avant le front C1 suivant de SCLK, le RECEPTEUR DE DONNEES positionne XONE à l'état faux puis surveille RONE après le front S1 suivant. C'est le bit de détection de désynchronisation.
- 11) Si le bit 1 d'état de trame est un bit_zéro et/ou le bit de détection de désynchronisation est un bit_un, il y a un problème avec la trame. Dans ce cas, le RECEPTEUR DE DONNEES retourne simplement à l'état de repos.
- 12) (Le bit 1 d'état de trame est un bit_un et le bit de détection de désynchronisation est un bit_zéro.) Le transfert de données du bus série est réussi. Sur le temps suivant de l'horloge C1, le RECEPTEUR DE DONNEES transfère la donnée reçue du registre de maintien de données vers le registre de données disponibles à travers le port de données, transfère le code mémorisé de type de trame vers le registre disponible à travers le port de dimension de données et retourne à l'état de repos. La sortie R STROBE du RECEPTEUR D'EN-TETE apparié avertit les couches supérieures du gestionnaire du bus série que des données nouvelles sont arrivées.

- 6) If the Frame Type code indicates a length the DATA RECEIVER can handle, it monitors and stores the indicated number of bytes from RONE, after each following S1 edge of SCLK. It stores the data in a Data Holding register, starting with the most significant bit of the most significant (leftmost) byte and ending with the least significant bit of the least significant (rightmost) byte.

PERMISSION 5.7:

In a Transaction Listener implementation, which is designed to provide data to higher layers "once and only once", as described in Paragraphs 5.5.5 and 6.1.6, designers MAY omit the Data Holding register and MAY store data directly into the Data register which is available via the Data Port.

OBSERVATION 5.19:

In a Transaction Listener, if there are subsequent problems in the Status subframe, the R STROBE output of the paired HEADER RECEIVER is simply not pulsed, and thus higher layers are not signalled that new data has arrived.

- 7) After the DATA RECEIVER has stored the last data bit, it monitors RONE after the next S1 edge of SCLK. This is bit 2 of the Frame Status field. If it monitors RONE True, there are problems with the frame, and the DATA RECEIVER simply returns to idle state.
- 8) (Bit 2 of the Frame Status is a zero_bit.) After the next S1 edge of SCLK the DATA RECEIVER again monitors RONE (bit 1 of the Frame Status field) and saves its state.
- 9) Before the next C1 edge of SCLK, the DATA RECEIVER makes XONE True to show that it is selected. This is bit 0 of the Frame Status field.
- 10) Before the next C1 edge of SCLK, the DATA RECEIVER makes XONE False and then monitors RONE after the following S1 edge. This is the Jam Detect bit.
- 11) If bit 1 of the Frame Status is a zero_bit and/or the Jam Detect bit is a one_bit, there is a problem with the frame. In this case the DATA RECEIVER simply returns to idle state.
- 12) (Bit 1 of the Frame Status is a one_bit and the Jam Detect bit is a zero_bit.) The serial bus data transfer is successful. On the following C1 clock, the DATA RECEIVER transfers the received data from the Data Holding register to the Data register which is available via Data Port, transfers the stored Frame Type code to the register that is available via Data Size Port and returns to idle state. The R STROBE output of the paired HEADER RECEIVER signals higher layers of serial bus management that new data has arrived.

5.6 GESTIONNAIRE DE TRAME

Le GESTIONNAIRE DE TRAME "referme la boucle" des modules du bus série qui sont décrits ici, en rapportant le résultat de la transmission de la trame vers les couches supérieures du gestionnaire du bus série, sur la carte de l'EMETTEUR D'EN-TETE qui a déclenché la trame.

OBSERVATION 5.20:

Un GESTIONNAIRE DE TRAME est normalement apparié avec un EMETTEUR D'EN-TETE, mais pourrait être autonome pour surveiller tout le trafic du bus série.

OBSERVATION 5.21:

Puisque la sortie SELECT de l'EMETTEUR D'EN-TETE et les sorties S et R SELECT d'un RECEPTEUR D'EN-TETE ont la même chronologie, un GESTIONNAIRE DE TRAME pourrait être apparié aussi bien avec un RECEPTEUR D'EN-TETE, mais une telle combinaison n'a pas une utilisation évidente.

Le GESTIONNAIRE DE TRAME assure deux autres fonctions importantes. Premièrement, il "suit" la transmission de chaque trame sur le bus série et informe l'EMETTEUR D'EN-TETE avec lequel il est apparié lorsque le bus est libre pour une trame nouvelle.

Deuxièmement, il guette en permanence les bit_départ: s'il observe un bit_départ *lorsqu'il suit une trame*, une erreur due au bruit sur le bus série a été la cause de la "désynchronisation de trame" avec un ou plusieurs autres GESTIONNAIRES DE TRAME du système. Dans ce cas il "désynchronise" le bus série en envoyant une chaîne de 512 bit_un. Cela invalide la trame en cours: la trame est ignorée par tous les modules du bus série. Il remet ainsi tous les autres GESTIONNAIRES DE TRAME et EMETTEUR D'EN-TETE du bus série en synchronisme de trame.

5.6.1 Interface avec la couche physique

Comme le montre la figure 5-1, page 112, l'interface avec la couche physique d'un GESTIONNAIRE DE TRAME inclut les signaux SCLK, XONE, XJAM, RONE et RSTART.

5.6.2 Interface avec la couche de liaison de données

L'interface avec la couche de liaison de données du GESTIONNAIRE DE TRAME, présentée dans la figure 5-1, inclut les signaux suivants:

SELECT

Un signal d'entrée de l'EMETTEUR D'EN-TETE apparié au GESTIONNAIRE DE TRAME, utilisé pour indiquer qu'il a obtenu l'arbitrage du bus série et a déclenché la trame en cours. Cette entrée valide les sorties SENT et CANCELLED pour la trame.

5.6 FRAME MONITOR module

The FRAME MONITOR "closes the loop" of the serial bus modules we have been describing, by reporting the result of a frame transmission back to higher layers of serial bus management, on the board with the HEADER SENDER which initiated the frame.

OBSERVATION 5.20:

A FRAME MONITOR is normally paired with a HEADER SENDER, but could stand alone to monitor all serial bus traffic.

OBSERVATION 5.21:

Since the SELECT output of a HEADER SENDER and the S and R SELECT outputs of a HEADER RECEIVER have the same timing, a FRAME MONITOR could be paired with a HEADER RECEIVER as well, but this combination has no obvious use.

The FRAME MONITOR provides two other important functions. First, it "tracks" the transmission of every frame on the serial bus and informs its paired HEADER SENDER when the bus is free for a new frame.

Second, it watches for start_bits at all times: if it sees a start_bit while it is tracking a frame, an error caused by noise on the serial bus has caused one or more of the other FRAME MONITORS in the system to be "out of frame synchronization" with this one. In this case, it "jams" the serial bus by sending a string of 512 one_bits. This invalidates the current frame: the frame is ignored by all serial bus modules. It also brings all FRAME MONITORS and HEADER SENDERS on the serial bus back into frame synchronization.

5.6.1 Physical Layer interface

As shown in Figure 5-1, page 113, the Physical Layer interface of a FRAME MONITOR includes the SCLK, XONE, XJAM, RONE and RSTART signals.

5.6.2 Link Layer interface

As shown in Figure 5-1, the Link Layer interface of a FRAME MONITOR includes the following signals:

SELECT

An input from the paired HEADER SENDER, used to signal that the HEADER SENDER has won the arbitration for the serial bus and has initiated the current frame. This input enables the SENT and CANCELLED outputs for the frame.

OBSERVATION 5.22:

Pour un GESTIONNAIRE DE TRAME qui n'est pas apparié à un EMETTEUR D'EN-TETE, il convient que cette entrée soit en permanence à l'état vrai.

FRAME IN PROGRESS Une sortie vers un EMETTEUR D'EN-TETE apparié. Lorsque cette sortie est à l'état faux, le bus série est libre et l'EMETTEUR D'EN-TETE peut déclencher une trame.

5.6.3 Interface avec les couches de niveau supérieur

L'interface du GESTIONNAIRE DE TRAME avec les couches de niveau supérieur, présentée dans la figure 5-1, page 112, inclut les signaux suivants:

RESET*	Lorsque ce signal est valide, le GESTIONNAIRE DE TRAME est initialisé comme décrit dans le paragraphe suivant.
Port de priorité	Trois signaux de sortie parallèles et les signaux de commande associés, par lesquels le champ de priorité des trames du bus série est communiqué aux couches supérieures du gestionnaire du bus série.
Port de code S, R	Deux groupe de signaux, chacun avec 10 sorties parallèles et les signaux de contrôle associés, par lesquels les adresses S et R des trames du bus série sont communiquées aux couches supérieures du gestionnaire du bus série.
Port de trame de données	Un signal de sortie indiquant si la trame communiquée aux couches supérieures de données était une trame de transfert de données. Cette information est nécessaire pour l'interprétation de l'information à partir du port de mot d'état.
Port de mot d'état	Trois signaux de sortie parallèles et les signaux de commande associés, par lesquels le champ du mot d'état de trame des trames du bus série est communiqué aux couches supérieures du gestionnaire du bus série.

AUTORISATION 5.8:

Les ports ci-dessus PEUVENT être réalisés comme des signaux de sorties dédiées ou comme un interface commun au bus.

OBSERVATION 5.22:

For a FRAME MONITOR which is not paired with a HEADER SENDER, this input should be permanently True.

FRAME IN PROGRESS An output to a paired HEADER SENDER. When this output is False, the serial bus is free and the HEADER SENDER can initiate a frame.

5.6.3 Higher Layer Interface

As shown in Figure 5-1, page 113, the Higher Layer interface of a FRAME MONITOR includes the following signals:

RESET*	When this signal is asserted, the FRAME MONITOR is initialized as described in the following paragraph.
Priority Port	Three parallel outputs and associated control signals, whereby the Priority field of serial bus frames is reported to higher layers of serial bus management.
S, R Code Ports	Two sets of signals, each with 10 parallel outputs and associated control signals, whereby the S and R addresses of serial bus frames are reported to higher layers of serial bus management.
Data Frame Port	An output signal indicating whether the frame being reported to higher layers was a Data Transfer frame. This information is needed to interpret the information from the Status port.
Status Port	Three parallel outputs and associated control signals, whereby the Frame Status field of serial bus frames is reported to higher layers of serial bus management.

PERMISSION 5.8:

The above ports MAY be implemented as dedicated outputs or as a common bus interface.

SENT

Un signal de sortie qui indique qu'une trame déclenchée par l'EMETTEUR D'EN-TETE apparié a été terminée et que les résultats de la transmission de la trame sont disponibles sur les cinq ports ci-dessus. Ce signal de sortie est à l'état vrai seulement si l'EMETTEUR D'EN-TETE couplé a signalé par le signal SELECT qu'il avait déclenché une trame. Il n'est pas positionné à l'état vrai pour une trame annulée ou pour une trame désynchronisée.

CANCELLED

Une variante au signal de sortie SENT. Il indique qu'une trame déclenchée par l'EMETTEUR D'EN-TETE couplé a été annulée par le RECEPTEUR D'EN-TETE qui n'est "pas prêt". Seulement les ports de priorité et de code S et R sont validés pour une trame annulée. Ce signal de sortie n'est pas positionné à l'état vrai si l'EMETTEUR D'EN-TETE couplé n'a pas activé sa sortie SELECT, ainsi qu'en cas de trame désynchronisée.

SUGGESTION 5.3:

Concevoir les groupes de gestionnaires qui diminuent leur niveau de priorité lors des essais ultérieurs (par exemple à 000) chaque fois qu'ils déclenchent une trame et qu'elle est annulée.

OBSERVATION 5.23:

Diminuer le niveau de priorité de cette façon permet aux autres EMETTEURS D'EN-TETE d'envoyer des trames sur le bus série. Cette stratégie, en plus de l'augmentation du niveau de priorité lorsqu'un EMETTEUR D'EN-TETE perd l'arbitrage du bus série (voir paragraphe 5.2.3), aide à assurer que chaque EMETTEUR D'EN-TETE obtienne un partage plus équitable de l'allocation du bus.

5.6.4 Initialisation

Chaque fois que RESET* est au niveau bas, le GESTIONNAIRE DE TRAME s'initialise comme suit:

- 1) Il positionne ses signaux de sortie SENT et CANCELLED à l'état faux.
- 2) Il positionne son signal de sortie FRAME IN PROGRESS à l'état vrai.
- 3) Il positionne XONE à l'état vrai et ignore RONE comme signal d'entrée.

Lorsque RESET* passe au niveau haut, le GESTIONNAIRE DE TRAME positionne XONE et FRAME IN PROGRESS à l'état faux et entre dans l'état de repos, décrit ci-dessous.

SENT

An output which indicates that a frame initiated by the paired HEADER SENDER has been completed and that the results of the frame transmission are available on the above five ports. This output is made True only if the paired HEADER SENDER has signalled that it initiated the frame, by means of the SELECT signal. It is not made True for a Cancelled frame or for a jammed frame.

CANCELLED

An alternative output to SENT. It indicates that a frame initiated by the paired HEADER SENDER has been cancelled by a HEADER RECEIVER which is "not ready". Only the Priority and S and R Code ports are valid for a Cancelled frame. This output is not made True if the paired HEADER SENDER has not signalled on SELECT, nor for a jammed frame.

SUGGESTION 5.3:

Design Controller groups to reduce the Priority value for subsequent retries (e.g. to 000) whenever they initiate a frame and it is cancelled.

OBSERVATION 5.23:

Reducing the Priority value in this way helps allow other HEADER SENDERS to send frames on the serial bus. This strategy, plus increasing the Priority value when a HEADER SENDER loses serial bus arbitration (see Paragraph 5.2.3), helps ensure that each HEADER SENDER gets a fair share of access to the bus.

5.6.4 Initialization

Whenever RESET* is Low, a FRAME MONITOR initializes itself as follows:

- 1) It makes its SENT and CANCELLED outputs False.
- 2) It makes its FRAME IN PROGRESS output True.
- 3) It makes XONE True and ignores RONE as an input.

When RESET* goes high, the FRAME MONITOR makes XONE and FRAME IN PROGRESS False and enters idle state, as described below.

5.6.5 Fonctionnement

REGLE 5.9:

Après avoir effectué la procédure d'initialisation ci-dessus, un GESTIONNAIRE DE TRAME DOIT opérer comme défini par le diagramme d'état dans la figure 5-9, page xx, et la description narrative suivante.

En partant de l'état de repos, le GESTIONNAIRE DE TRAME opère ainsi:

- 1) Il surveille RSTART dans chaque cycle SCLK jusqu'à la détection de RSTART à l'état vrai après un front S2. Cela indique un bit_départ.

Sur le premier front C1 après le retour à partir des étapes 6 ou 10 ci-après, le GESTIONNAIRE DE TRAME positionne ses signaux de sortie SENT et CANCELLED au niveau faux.

- 2) Après le front C1 suivant la détection de bit_départ et avant le front S2 suivant, il positionne son signal de sortie FRAME IN PROGRESS à l'état vrai.
- 3) Commençant après le front S1 qui suit le front S2 sur lequel a été détecté le bit_départ, le GESTIONNAIRE DE TRAME surveille les bits successifs sur RONE après les fronts S1 de SCLK et les range de la manière suivante:

3 bits après le bit_départ: mémorisés dans le port de priorité

Les 10 bits suivants: mémorisés dans le port de code S

Les 10 bits suivants: mémorisés dans le port de code R

Le bit suivant: rejeté/ignoré

- 4) Pour chacun des bits de l'étape 3 et pour les étapes suivantes jusqu'au retour à l'état de repos, le GESTIONNAIRE DE TRAME surveille aussi RSTART après chaque front S2. S'il détecte RSTART à l'état vrai après S2, il positionne son signal de sortie XJAM à l'état vrai entre les fronts C1 et S1 suivants. Cela induit le module d'ACCES AU BUS à positionner le signal SERDAT* au niveau bas pour un bit_un d'une manière combinatoire (immédiate). Avant le front C1 suivant, il positionne XONE à l'état vrai et après ce front C1, il positionne XJAM à l'état faux. Après, il maintient XONE vrai pendant au moins 510 autres fronts C1 de SCLK. Avant le front C1 suivant cette durée, il positionne XONE à l'état faux et revient à l'état de repos.

OBSERVATION 5.24:

Le signal XJAM est inclus dans l'interface couche physique/couche de liaison des données parce qu'il est peu probable qu'un GESTIONNAIRE DE TRAME puisse positionner XONE vrai assez rapidement (en réponse à RSTART) afin que le module d'ACCES AU BUS puisse l'échantillonner de façon fiable sur le front C1.

5.6.5 Operation

RULE 5.9:

After completing the above initialization procedure, a FRAME MONITOR MUST operate as defined by the state diagram in Figure 5-9, page 171, and the following narrative description.

Starting from idle state, a FRAME MONITOR proceeds as follows:

- 1) It monitors RSTART in every SCLK cycle until it monitors RSTART True after an S2 transition. This indicates a start_bit.

On the first C1 edge after returning from step 6 or 10 below, the FRAME MONITOR makes its SENT and CANCELLED outputs False.

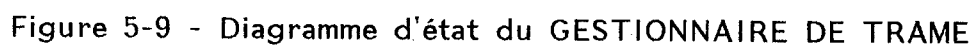
- 2) After the C1 edge after monitoring a start_bit, and before the next S2 edge, it makes its FRAME IN PROGRESS output True.
- 3) Starting after the S1 edge that follows the S2 edge on which it found a start_bit, the FRAME MONITOR monitors successive bits on RONE after S1 edges of SCLK and disposes of them as follows:

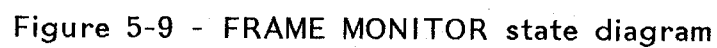
3 bits after start_bit: stored in Priority Port
Next 10 bits: stored in S Code Port
Next 10 bits: stored in R Code Port
Next bit: discarded/ignored

- 4) For each of the bits in step 3 and in subsequent steps until it returns to idle state, the FRAME MONITOR also monitors RSTART after each S2 edge. If it monitors RSTART True after S2, it makes its XJAM output True between the following C1 and S1 edges. This causes the BUS ACCESS module to make SERDAT* low for a one_bit, in a combinatorial (immediate) fashion. Before the next C1 edge, it makes XONE True and after the C1 edge, it makes XJAM False. It then maintains XONE True across at least 510 more C1 edges of SCLK. Before the next C1 edge thereafter, it makes XONE False and returns to idle state.

OBSERVATION 5.24:

The XJAM signal is included in the Physical/Link Layer interface because it is unlikely that a FRAME MONITOR could make XONE True fast enough (in response to RSTART) so that the BUS ACCESS module could reliably sample it on the C1 edge.





- 5) Ensuite, le GESTIONNAIRE DE TRAME échantillonne son signal d'entrée SELECT sur le front S1 suivant et retient le résultat pour plus tard. Après ce front, il surveille le bit suivant sur RONE, le bit HSVAL. S'il détecte RONE à l'état faux, alors il positionne FRAME IN PROGRESS à l'état faux sur le front C1 suivant, et (à moins d'avoir détecté RSTART à l'état vrai après S2) il revient à l'état de repos.
 - 6) (HSVAL a été un bit_un.) Le GESTIONNAIRE DE TRAME surveille alors les trois bits suivants sur RONE (et RSTART). Ceux-ci représentent le code de type de trame.
 - 7) Si le type de trame est 111, il positionne FRAME IN PROGRESS à l'état faux sur le front C1 suivant. Après le front S2 suivant, il surveille RONE et RSTART (c'est le bit de détection de désynchronisation). Si le bit de détection de désynchronisation est un bit_zéro et le signal SELECT a été échantillonné à l'état vrai pendant l'étape 5, le GESTIONNAIRE DE TRAME positionne CANCELLED à l'état vrai sur le front C1 suivant. Excepté la détection d'un bit_départ, il retourne à l'état de repos pour surveiller le bit suivant le bit de détection de désynchronisation. (A l'état de repos, il va positionner CANCELLED à l'état faux sur le front C1 suivant.)
 - 8) Si le type de trame est 001-110, le GESTIONNAIRE DE TRAME traduit ce code en le nombre de bits de données dans la trame, comme indiqué précédemment dans ce chapitre. Sur le front C1 suivant, il positionne le bit du registre du port de trame de donnée à l'état vrai et puis surveille RONE et RSTART pour ce nombre de bits de données. Durant la transmission des bits de données, le GESTIONNAIRE DE TRAME surveille les bit_départ comme décrit à l'étape 4, tout en rejetant les données. Il procède ensuite à l'étape 10.
 - 9) Si le type de trame est 000, le GESTIONNAIRE DE TRAME positionne le bit du registre du port de trame de données à l'état faux sur le front C1 suivant.
 - 10) Après chacun des trois fronts S1 suivants après les bits de donnée (s'ils existent, sinon après les bits de type de trame), le GESTIONNAIRE DE SURVEILLANCE DE TRAME surveille RONE (et RSTART) et mémorise les bits dans le port de mot d'état.
 - 11) Sur le front C1 après le dernier bit de mot d'état, le GESTIONNAIRE DE TRAME positionne FRAME IN PROGRESS à l'état faux. Après le front S2 suivant, il surveille RONE et RSTART (c'est le bit de détection de désynchronisation). Si le bit de détection de désynchronisation est un bit_zéro et SELECT a été échantillonné à l'état vrai à l'étape 5, il positionne SENT à l'état vrai sur le front C1 suivant. Excepté la détection d'un bit_départ, il retourne à l'état de repos pour surveiller le bit suivant le bit de détection de désynchronisation. (Dans l'état de repos il positionnera SENT à l'état faux sur le front C1 suivant.)
-

- 5) The FRAME MONITOR then samples its SELECT input on the next S1 edge and retains the result for later. After that edge it monitors the next bit on RONE, the HSVAL bit. If it monitors RONE False it makes FRAME IN PROGRESS False at the following C1 edge, and (unless it monitors RSTART True after S2) it returns to idle state.
 - 6) (HSVAL was a one_bit.) The FRAME MONITOR then monitors the next three bits on RONE (and RSTART). These are the Frame Type code.
 - 7) If the Frame Type is 111, it makes FRAME IN PROGRESS False on the following C1 edge. After the following S2 edge, it monitors RONE and RSTART (this is the Jam Detect bit). If the Jam Detect bit is a zero_bit and SELECT was sampled True in step 5, the FRAME MONITOR makes CANCELLED True on the following C1 edge. Unless it monitored a start_bit, it then returns to idle state to monitor the bit following the Jam Detect bit. (In idle state it will make CANCELLED False on the next C1 edge.)
 - 8) If the Frame Type is 001-110, the FRAME MONITOR translates this code into the number of Data bits in the frame, as described earlier in this chapter. On the next C1 edge, it makes the register bit for the Data Frame Port True and then monitors RONE and RSTART for that number of Data bits. During the Data bits the FRAME MONITOR monitors for start_bits as described in step 4, but otherwise discards the data. It then proceeds to step 10.
 - 9) If the Frame Type is 000, on the next C1 edge the FRAME MONITOR makes the register bit for the Data Frame Port False.
 - 10) After each of the next three S1 edges after the Data bits (if present, else after the Frame Type bits), the FRAME MONITOR monitors RONE (and RSTART) and stores the bits in the Status port.
 - 11) On the C1 edge after the last Status bit, the FRAME MONITOR makes FRAME IN PROGRESS False. After the following S2 edge it monitors RONE and RSTART (this is the Jam Detect bit). If the Jam Detect bit is a zero_bit and SELECT was sampled True in step 5, it makes SENT True on the following C1 edge. Unless it monitored a start_bit, it then returns to idle state to monitor the bit following the Jam Detect bit. (In idle state it will make SENT False on the next C1 edge.)
-

CHAPITRE 6: GROUPES DE LA COUCHE DE LIAISON DE DONNEES ET PROTOCOLE

Le chapitre précédent a défini les cinq types de modules de la couche de liaison de données. Ce chapitre décrit comment ces modules peuvent être combinés en groupes qui interagissent sur le bus série afin d'accomplir des tâches utiles.

AUTORISATION 6.1:

Les circuits du bus série PEUVENT être conçus pour utiliser les modules du bus série définis dans le chapitre 5, dans les groupes autres que ceux définis dans ce chapitre.

Au lieu d'utiliser les diagrammes d'état, ce chapitre décrit chaque groupe de modules par le moyen de schémas logiques qui montrent l'interconnexion de modules constituant un groupe, par les lignes de signaux et les éléments logiques communs. Pour chaque groupe il y a également une description orale de sa désignation et de son opération.

AUTORISATION 6.2:

Les cartes du bus série PEUVENT réaliser les fonctions de groupe décrites dans ce chapitre par le moyen d'une combinaison de logiques matérielle et logicielle.

6.1 Groupes simples

Cette section présente un nombre de moyens par lesquels les modules du bus série peuvent être combinés pour réaliser les groupes gestionnaires, parleurs, écouteurs et indicateurs avec différentes propriétés et protocoles fonctionnels.

6.1.1 Indicateur simple

C'est le type de groupe de modules du bus série le plus simple. Comme le présente la figure 6-1, page 176, il consiste en un RECEPTEUR D'EN-TETE avec ses deux entrées ENABLE S et ENABLE R mises à l'état vrai et son port de code initialisé à une valeur décimale comprise entre 1 et 1022. Les sorties STROBE S et STROBE R du RECEPTEUR D'EN-TETE sont connectées aux entrées de mise à un et de mise à zéro d'une bascule. Les sorties de la bascule forment la sortie de l'indicateur simple vers les couches supérieures du gestionnaire du bus série.

CHAPTER 6: LINK LAYER GROUPS AND PROTOCOL

The preceding chapter defined the five types of Link Layer modules. This chapter describes how these modules can be combined into groups that interact on the serial bus to perform useful tasks.

PERMISSION 6.1:

Serial bus circuits MAY be designed to use the serial bus modules defined in Chapter 5, in groups other than those defined in this chapter.

Instead of using state diagrams, this chapter describes each module group by means of logic diagrams which show the group's constituent modules interconnected by signal lines and common logic elements. For each group there is also a verbal description of its purpose and operation.

PERMISSION 6.2:

Serial bus boards MAY implement the group functions described in this chapter by means of a mixture of hardware and software logic.

6.1 Simple groups

This section presents a number of ways in which serial bus modules can be combined to make Controller, Talker, Listener and Flag groups with various properties and operational protocols.

6.1.1 Simple Flag

This is the simplest type of serial bus module group. As shown in Figure 6-1, page 177, it consists of a HEADER RECEIVER with its ENABLE S and ENABLE R inputs both set to True, and its Code Port set to some decimal value in the range 1-1022. The S STROBE and R STROBE outputs of the HEADER RECEIVER are connected to the Set and Reset inputs of a latch. The outputs of the latch form the output of the Simple Flag to higher layers of serial bus management.

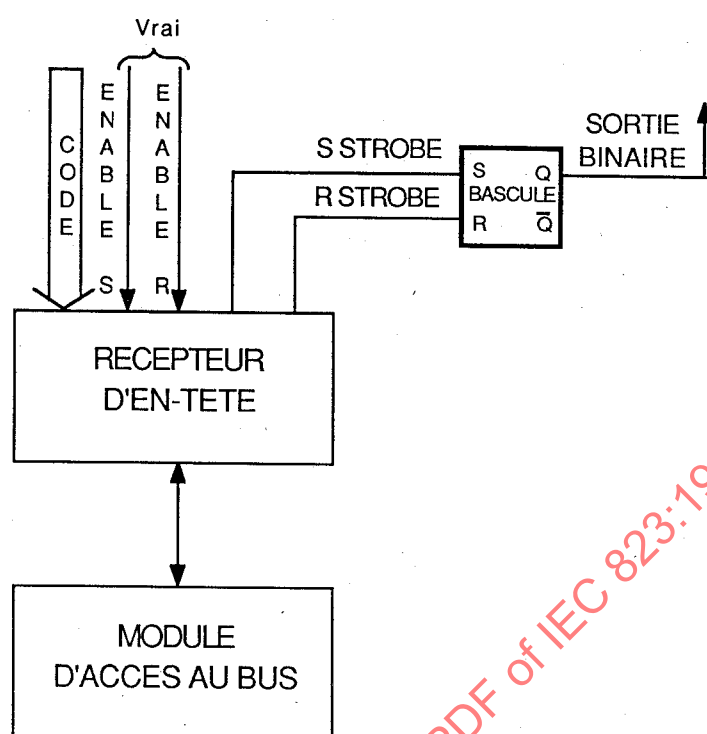


Figure 6-1 - Indicateur simple

6.1.1.1 But

Un indicateur simple peut être utilisé partout où une sortie de signal discret ou un bit d'état enregistré est nécessaire dans un système de bus série. Une application pour l'indicateur simple est d'offrir un signal qui peut valider/invalider l'interface au bus système d'une carte ou un signal pour remettre à zéro les circuits logiques d'une carte. Un indicateur simple peut également être utilisé dans les applications de signal virtuel, comme décrit au paragraphe 6.1.2.

6.1.1.2 Fonctionnement

Le RECEPTEUR D'EN-TETE dans ce groupe compare la valeur provenant de son port de code avec les adresses S et R dans chaque en-tête sur le bus série. Si elle ne correspond à aucune des deux adresses, il ignore la trame.

Puisque les entrées ENABLE S et ENABLE R du RECEPTEUR D'EN-TETE sont toutes les deux à l'état vrai, un groupe d'indicateurs simples n'annule jamais une trame. A la place, s'il ne trouve pas une correspondance, le RECEPTEUR D'EN-TETE effectue ensuite une surveillance du champ de type de trame de la trame. S'il trouve la valeur 111, indiquant qu'un autre module a annulé la trame, il ignore la trame. Autrement, pour une trame de contrôle ou une trame de transfert de données, il poursuit la surveillance au moins jusqu'au bit le plus significatif (MSB) du champ du mot d'état de la trame et au bit de détection de désynchronisation. Si le MSB du mot d'état de la trame est 1, ou si le bit de détection de désynchronisation est à 1, le RECEPTEUR D'EN-TETE ignore la trame d'une façon similaire.

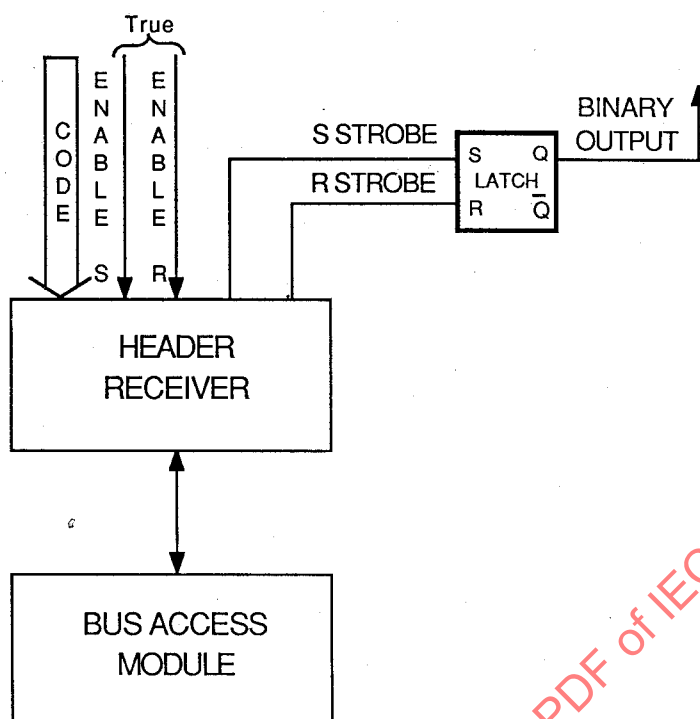


Figure 6-1 - Simple Flag

6.1.1.1 Purpose

A Simple Flag can be used wherever a discrete signal output or registered status bit is needed in a serial bus system. One application of a Simple Flag is to provide a signal that can enable/disable the system bus interface of a board, or a signal to Reset the logic of a board. A Simple Flag can also be used in Virtual Signal applications, as described in Paragraph 6.1.2.

6.1.1.2 Operation

The HEADER RECEIVER in this group compares the value from its Code port to the S and R addresses in each Header on the serial bus. If it does not find a match in either of these addresses, it ignores the frame.

Since the ENABLE S and ENABLE R inputs of the HEADER RECEIVER are both True, a Simple Flag group never cancels a frame. Instead, if it does find a match, the HEADER RECEIVER then monitors the Frame Type field of the frame. If it monitors the value 111, indicating that some other module cancelled the frame, it ignores the frame. Otherwise, for a Control frame or Data Transfer frame, it goes on to monitor at least the Most Significant Bit (MSB) of the Frame Status field and the Jam Detect bit. If the MSB of the Frame Status is 1 or the Jam Detect bit is 1, the HEADER RECEIVER similarly ignores the frame.

Pour une trame de transfert de données, il y a une vérification de plus: le RECEPTEUR D'EN-TETE surveille les deux bits les moins significatifs du mot d'état de la trame et, s'ils ne sont pas 11, il ignore la trame. Dans une trame de commande, il présente un 1 sur un de ces deux bits, selon l'adresse S ou R, et il ignore l'autre bit.

En résumé des vérifications décrites ci-dessus, le RECEPTEUR D'EN-TETE ignore les trames qui n'ont pas abouti. Pour une trame qui aboutit, s'il y a correspondance avec l'adresse S, il active sa sortie STROBE S à l'état vrai, positionnant sa bascule. Pour une trame réussie dans laquelle il y a correspondance avec l'adresse R, le RECEPTEUR D'EN-TETE active sa sortie STROBE R à l'état vrai, remettant à zéro sa bascule.

6.1.2 Gestionnaire de signal virtuel

La figure 6-2 ci-dessous représente un gestionnaire de signal virtuel. Comme les autres groupes de gestionnaires, il comprend un EMETTEUR D'EN-TETE et un GESTIONNAIRE DE TRAME. Il diffère des autres gestionnaires par le fait que son entrée SEND 12 est commandée par un signal physique discret ENTREE BINAIRE, et sa ligne SEND 21 est commandée par ce même signal inversé.

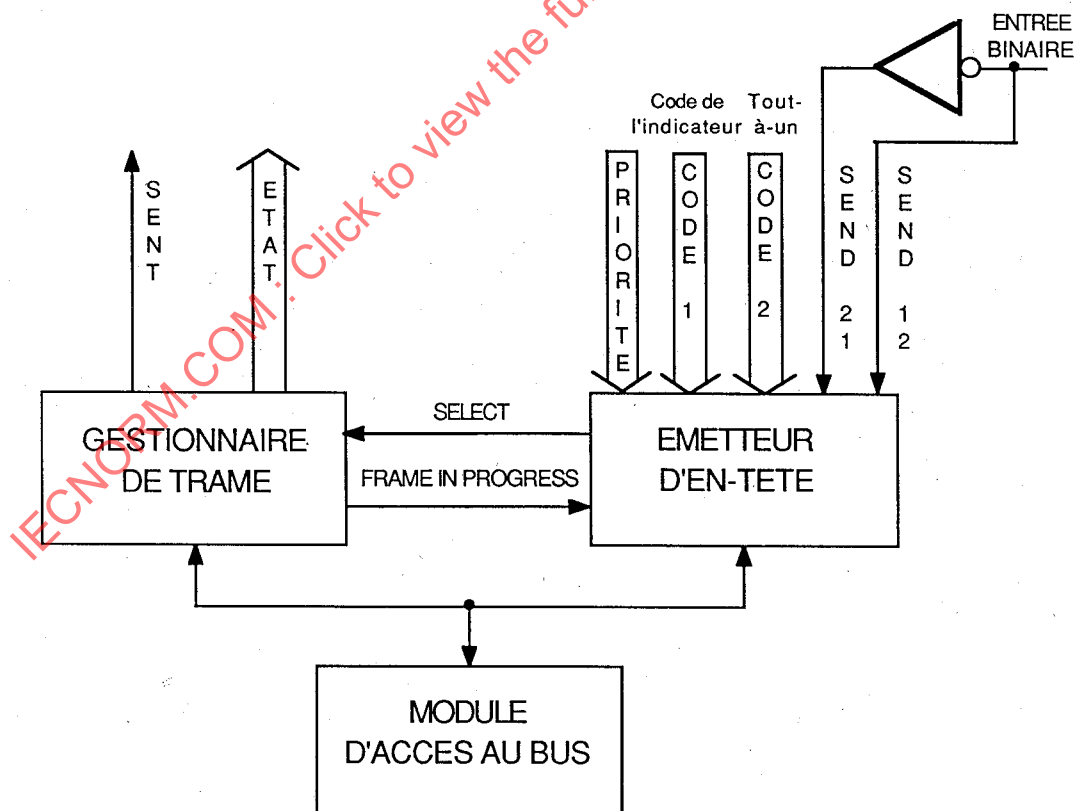


Figure 6-2 - Gestionnaire de signal virtuel

For a Data Transfer frame there is one further check: the HEADER RECEIVER also monitors the two less-significant bits of the Frame Status and, if they are not 11, it ignores the frame. In a Control frame, it presents a 1 on one of these bits, depending on whether it matched the S or the R address, and ignores the other bit.

The summary of the above checks is that the HEADER RECEIVER ignores frames that are not successful. For a successful frame, if it matched the S address, it pulses its S STROBE output True, setting its latch. For a successful frame in which it matched the R address, the HEADER RECEIVER pulses its R STROBE output True, resetting its latch.

6.1.2 Virtual Signal Controller

Figure 6-2 shows a Virtual Signal Controller. Like other Controller groups, it is composed of a HEADER SENDER and a FRAME MONITOR. It differs from other Controllers in that its SEND 12 input is controlled by a discrete physical signal BINARY INPUT, and its SEND 21 line is controlled by the inversion of the same signal.

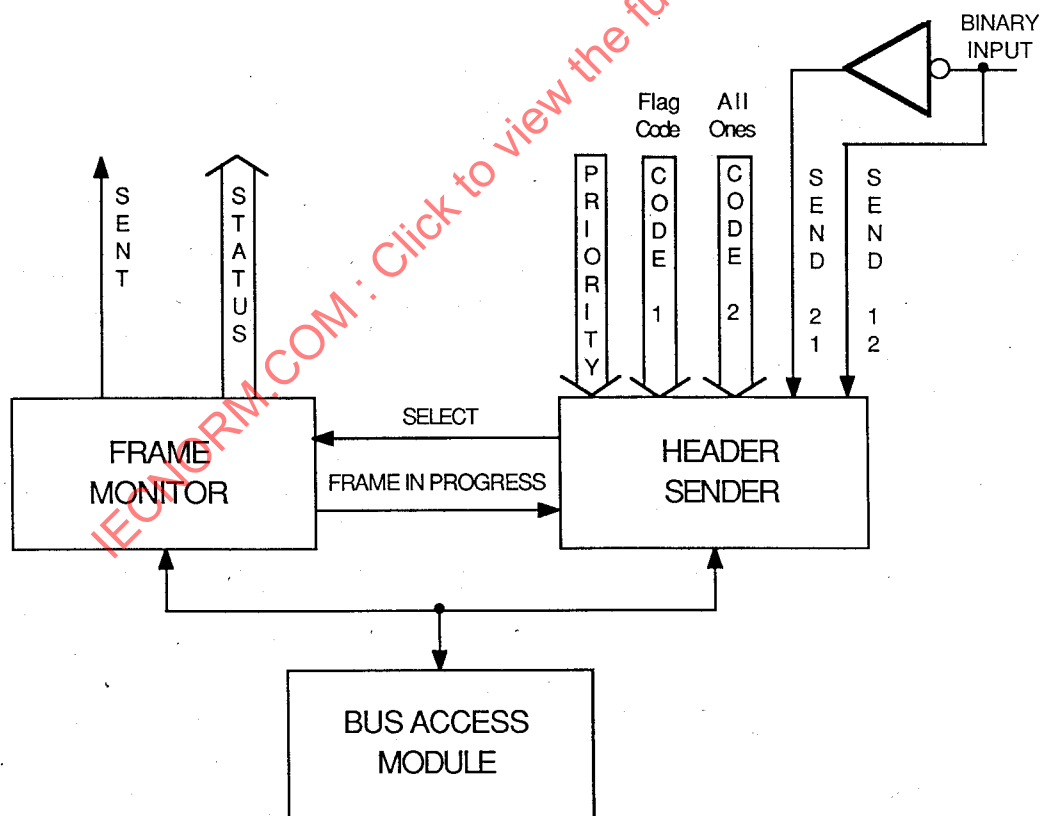


Figure 6-2 - Virtual Signal Controller

6.1.2.1 But

Un gestionnaire de signal virtuel sur une carte peut être utilisé avec un module d'indicateur simple sur une autre carte pour fournir une fonction de signal virtuel. Dans cette implantation, un signal de sortie de l'indicateur simple suit l'état d'un signal d'entrée et indique son état au gestionnaire de signal virtuel avec un "temps de propagation" qui est proportionnel au taux du trafic total sur le bus série.

6.1.2.2 Fonctionnement

Une configuration avec signal virtuel inclut un gestionnaire de signal virtuel et un indicateur simple qui ont été fournis avec les entrées suivantes au moment de l'initialisation et de la mise en fonctionnement du bus série:

- 1) Le port de priorité du gestionnaire est initialisé à une valeur appropriée.
- 2) Le port de code 1 du gestionnaire et le port de code de l'indicateur simple sont tous les deux initialisés à la même valeur comprise entre 1 et 1022.
- 3) Le port de code 2 du gestionnaire est initialisé à "tout-à-un" qui ne sélectionne jamais un groupe du bus série.
- 4) Les entrées ENABLE S et ENABLE R de l'indicateur simple sont toutes les deux mises à l'état vrai.

REGLE 6.1:

Lorsqu'elles configurent un groupe gestionnaire pour envoyer un en-tête destiné *seulement* à positionner le groupe indicateur qui a été identifié par l'adresse S, les couches supérieures du gestionnaire du bus série DOIVENT assurer que l'adresse R ne sélectionne pas un groupe du bus série.

REGLE 6.2:

Lorsqu'elles configurent un groupe gestionnaire pour envoyer un en-tête destiné *seulement* à mettre à zéro le groupe indicateur qui a été identifié par l'adresse R, les couches supérieures du gestionnaire du bus série DOIVENT assurer que l'adresse S ne sélectionne pas un groupe du bus série.

RECOMMANDATION 6.1:

Dans ces deux cas, s'il n'est pas important qu'un seul groupe gestionnaire obtienne l'arbitrage du bus série et émette son en-tête, utiliser "tout-à-un" dans les "autres" adresses.

Dès que les entrées ci-dessus sont initialisées comme indiqué, elles ne sont pas modifiées. Supposer, également, que ce gestionnaire soit le seul à émettre les en-têtes qui sélectionnent l'indicateur simple et qu'aucun autre type de module ne réponde à l'adresse de l'indicateur simple. (Bien sûr, les indicateurs simples sur plusieurs cartes différentes peuvent tous répondre à cette adresse.)

6.1.2.1 Purpose

A Virtual Signal Controller on one board can be used with a Simple Flag module on another board to provide a Virtual Signal function. In this arrangement, an output signal from the Simple Flag tracks and reflects the state of an input signal to the Virtual Signal Controller, with a "propagation time" that is proportional to the level of overall serial bus traffic.

6.1.2.2 Operation

A Virtual Signal configuration includes a Virtual Signal Controller and a Simple Flag which were provided with the following inputs at the time the serial bus system was initialized and set into operation:

- 1) The Controller's Priority Port was set to some appropriate value.
- 2) The Controller's Code 1 Port and the Simple Flag's Code Port were both set to the same value in the range 1-1022.
- 3) The Controller's Code 2 Port was set to "all ones" which never selects a serial bus group.
- 4) The Simple Flag's ENABLE S and ENABLE R inputs were both made True.

RULE 6.1:

When configuring a Controller group to send a Header which is intended *only* to set the Flag group identified by the S address, higher layers of serial bus management **MUST** ensure that the R address does not select a serial bus group.

RULE 6.2:

When configuring a Controller group to send a Header which is intended *only* to reset the Flag group identified by the R address, higher layers of serial bus management **MUST** ensure that the S address does not select a serial bus group.

RECOMMENDATION 6.1:

In either of the above cases, if it does not matter whether or not a single Controller group wins the serial bus arbitration and sends the Header, use "all-ones" in the "other" address.

Once the above inputs are set up as indicated, they are not changed. Suppose also that this Controller is the only one that sends Headers that select the Simple Flag and that no other type of module responds to the Simple Flag's address. (Of course, Simple Flags on several different boards can all respond to the address.)

Dès que le système est en fonctionnement, chaque fois que ENTREE BINAIRE passe à l'état vrai, SEND 12 passe à l'état vrai. Cela indique à l'EMETTEUR D'EN-TETE d'émettre un en-tête avec l'adresse de l'indicateur simple dans le champ S et "tout-à-un" dans l'adresse R. Quand l'EMETTEUR D'EN-TETE obtient l'arbitrage du bus série et émet l'en-tête, l'indicateur simple est sélectionné par l'adresse S et met à 1 sa bascule, mettant le signal SORTIE BINAIRE à l'état vrai.

Ensuite, l'EMETTEUR D'EN-TETE attend le passage à l'état faux de son entrée SEND 12 (c'est-à-dire le signal SORTIE BINAIRE). A ce moment, le signal SEND 21 passe à l'état vrai. Cela indique à l'EMETTEUR D'EN-TETE d'émettre un en-tête avec "tout-à-un" comme adresse S et l'adresse de l'indicateur simple dans le champ R. Quand l'EMETTEUR D'EN-TETE obtient l'arbitrage du bus série et émet l'en-tête, l'indicateur simple est sélectionné par l'adresse R et met à zéro sa bascule, qui met le signal SORTIE BINAIRE à l'état faux.

Le gestionnaire de signal virtuel et l'indicateur simple forment ainsi un circuit de signal virtuel dans lequel SORTIE BINAIRE indique les changements de ENTREE BINAIRE (après un "temps de propagation"). La longueur de 10 bits des adresses S et R est la seule limitation sur le nombre de signaux virtuels qui peuvent être initialisés.

6.1.3 Parleur sur demande

Comme le montre la figure 6-3 ci-dessous, un parleur sur demande est constitué d'un RECEPTEUR D'EN-TETE et d'un EMETTEUR DE DONNEES, avec l'entrée ENABLE S du RECEPTEUR D'EN-TETE mise à l'état vrai. Dès que le parleur sur demande est initialisé et mis en fonctionnement, il est toujours valide pour émettre des données en réponse à un en-tête qui le sélectionne dans l'adresse S.

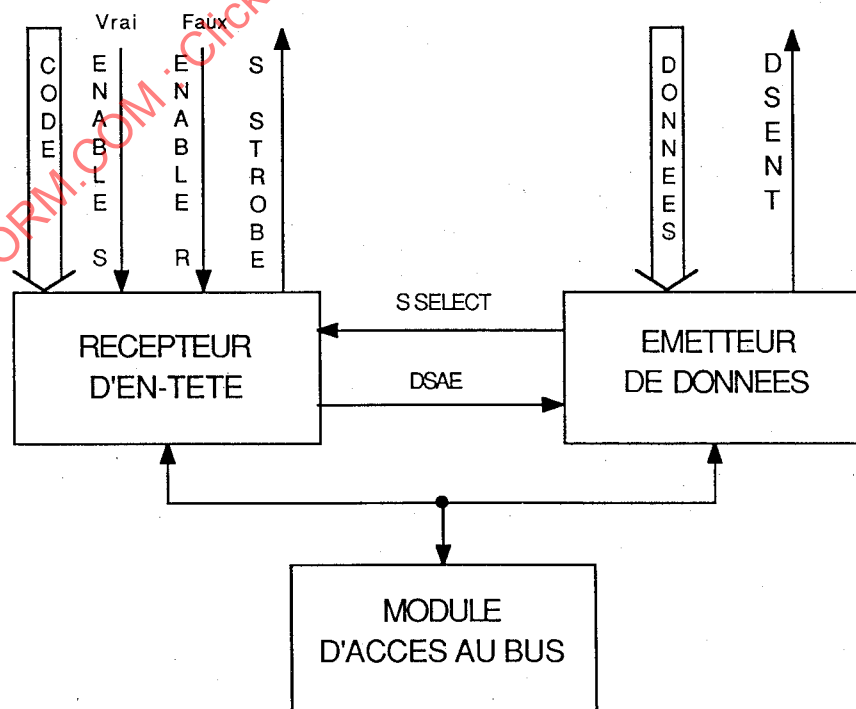


Figure 6-3 - Parleur sur demande

Once the system is in operation, whenever BINARY INPUT goes True, SEND 12 goes True. This signals the HEADER SENDER to send a Header with the Simple Flag's address in the S field and "all ones" in the R address. When the HEADER SENDER wins the serial bus arbitration and sends the Header, the Simple Flag is selected by the S address and sets its latch, making the BINARY OUTPUT signal True.

The HEADER SENDER then waits for its SEND 12 input (i.e. the BINARY INPUT signal) to go False. When this happens, the SEND 21 signal goes True. This signals the HEADER SENDER to send a Header with "all ones" as the S address and the Simple Flag's address in the R field. When the HEADER SENDER wins the serial bus arbitration and sends the Header, the Simple Flag is selected by the R address and clears its latch, making the BINARY OUTPUT signal False.

Thus the Virtual Signal Controller and the Simple Flag form a virtual signal circuit in which BINARY OUTPUT reflects changes in BINARY INPUT (after a "propagation time"). The 10 bit size of the S and R addresses is the only limit on how many such virtual signals can be set up.

6.1.3 On-Demand Talker

As shown in Figure 6-3, an On-Demand Talker is composed of a HEADER RECEIVER and a DATA SENDER, with the ENABLE S input of the HEADER RECEIVER set to True. Once the On-Demand Talker is initialized and set into operation, it is always enabled to send data in response to a Header that selects it in the S address.

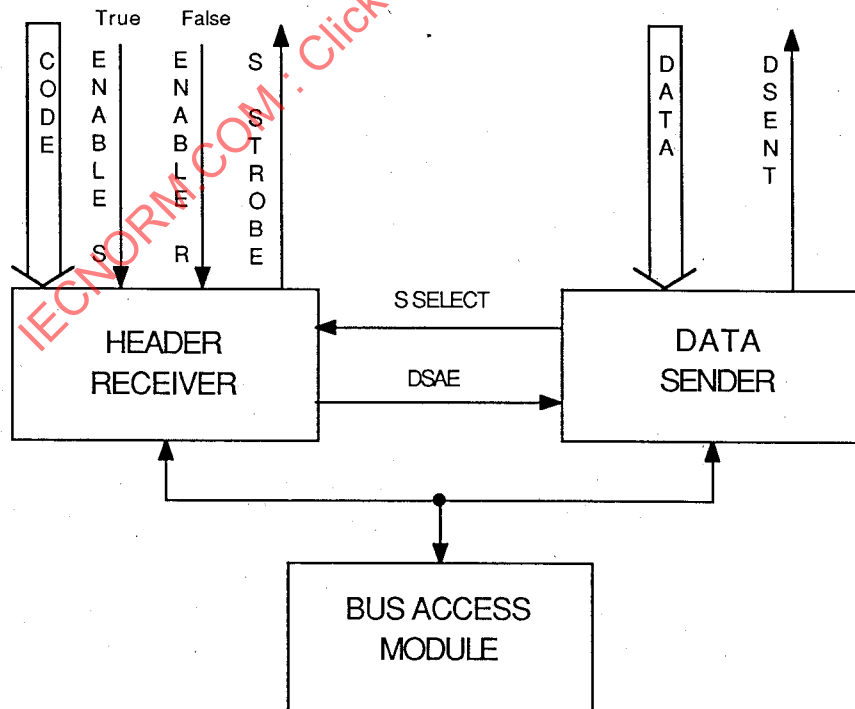


Figure 6-3 - On-Demand Talker

6.1.3.1 But

Un parleur sur demande peut être utilisé pour émettre des données vers les écouteurs sur d'autres cartes, dans des applications où les données sont émises quand elles sont nécessaires pour l'écouteur, au lieu de les envoyer lorsque les "nouvelles données" deviennent disponibles au parleur. Ainsi ce protocole est avantageux quand les données changent fréquemment, mais sont rarement demandées.

Plusieurs parleurs sur demande pourraient être utilisés avec un gestionnaire de lecture (voir paragraphe 6.2.2) pour envoyer la valeur la plus grande d'une variable critique, parmi les processus parallèles sous le contrôle des cartes du parleur, et cela commandé par un processeur de la carte du gestionnaire.

6.1.3.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un parleur sur demande de la façon suivante:

- 1) Elles positionnent le port de code du RECEPTEUR D'EN-TETE à une valeur comprise entre 1 et 1022.
- 2) Elles positionnent le port de code de l'EMETTEUR DE DONNEES aux données initiales à émettre.
- 3) Elles mettent ENABLE S à l'état vrai.
- 4) Elles mettent ENABLE R à l'état faux.

Périodiquement ensuite, les couches supérieures du gestionnaire du bus série présentent de nouvelles données au port de données de l'EMETTEUR DE DONNEES. Avec ce groupe, il n'y a pas de relation définie entre l'instant où une nouvelle donnée est chargée et celui où elle est envoyée sur le bus série.

REGLE 6.3:

Quand les couches supérieures du gestionnaire du bus série chargent de nouvelles données dans un parleur sur demande, le groupe NE DOIT PAS envoyer un mélange d'anciennes et de nouvelles données.

SUGGESTION 6.1:

Prévoir que les couches supérieures chargent les nouvelles données dans un "registre tampon" et puis signalent que le chargement est terminé. Le parleur sur demande transfère alors le registre tampon à son registre de transmission de données quand les couches supérieures ont signalé que le chargement est terminé ET qu'aucune trame n'est en cours sur le bus série.

Si un groupe gestionnaire, quelque part sur le bus série, envoie un en-tête avec l'adresse du parleur sur demande dans le champ S, le RECEPTEUR D'EN-TETE reconnaît l'adresse. Puisque ENABLE S est toujours vrai, le RECEPTEUR D'EN-TETE n'annule pas la trame, mais positionne plutôt S SELECT de l'EMETTEUR DE DONNEES.

6.1.3.1 Purpose

An On-Demand Talker can be used to send data to Listeners on other boards, in applications where the data is sent whenever it is needed at the Listener, rather than when "new data" becomes available at the Talker. Thus, this scheme is advantageous when data changes often but is seldom needed.

Several On-Demand Talkers might be used with a Reading Controller (see Paragraph 6.2.2) to send the largest value of a critical variable among parallel processes being controlled by the Talkers' boards, on command from a processor on the board with the Controller.

6.1.3.2 Operation

Higher layers of serial bus management configure an On-Demand Talker as follows:

- 1) They set the Code Port of the HEADER RECEIVER to some value in the range 1-1022.
- 2) They set the DATA SENDER's Data Port to the initial data to be sent.
- 3) They make ENABLE S True.
- 4) They make ENABLE R False.

Periodically thereafter, higher layers of serial bus management present new data at the DATA SENDER's Data Port. With this group, there is no defined relationship between when new data is loaded and when it is sent on the serial bus.

RULE 6.3:

When higher layers of serial bus management are loading new data into an On-Demand Talker, the group MUST NOT send a mixture of old and new data.

SUGGESTION 6.1:

Have higher layers load new data into a "holding register", and then signal that the loading process is complete. Have the On-Demand Talker transfer the holding register to its transmit data register whenever higher layers have signalled that loading is complete AND a frame is not in progress on the serial bus.

If a Controller group anywhere on the serial bus sends a Header with the On-Demand Talker's address in the S field, the HEADER RECEIVER recognizes the address. Since ENABLE S is always True, the HEADER RECEIVER does not cancel the frame, but instead asserts S SELECT to the DATA SENDER.

L'EMETTEUR DE DONNEES envoie alors l'une des valeurs 001-110 dans la sous-trame de type de trame selon le nombre d'octets qui ont été chargés dans le port de donnée par les couches supérieures du gestionnaire du bus série. L'EMETTEUR DE DONNEES utilise la logique d'arbitrage du bus série pendant qu'il envoie le type de trame. S'il perd l'arbitrage à la valeur 111, indiquant une trame annulée, l'EMETTEUR DE DONNEES et le RECEPTEUR D'EN-TETE ignorent tous deux la trame.

Si l'EMETTEUR DE DONNEES perd l'arbitrage du bus série dans le champ de type de trame à une valeur plus grande que celle de l'intervalle 010-110, il attend le nombre d'octets indiqué par la valeur gagnante et puis envoie 110 dans le champ d'état de trame pour indiquer un "conflit de dimension de données émetteur/émetteur".

Typiquement, l'EMETTEUR DE DONNEES gagne l'arbitrage du bus série dans le champ de type de trame et puis envoie le nombre d'octets indiqué immédiatement après. Il peut ou non utiliser la logique d'arbitrage du bus série pendant qu'il envoie les données, selon le bit DSAE de l'en-tête qui est répété dans le signal DSAE de l'EMETTEUR D'EN-TETE vers l'EMETTEUR DE DONNEES.

Si l'EMETTEUR DE DONNEES utilise la logique d'arbitrage du bus série et perd l'arbitrage dans le champ de données, il ignore le reste de la trame. Autrement, après les données, il surveille le bit le plus significatif suivant du mot d'état de la trame. Si le MSB est à 1, l'EMETTEUR DE DONNEES et le RECEPTEUR D'EN-TETE ignorent le reste de la trame. Si le MSB est à 0, l'EMETTEUR DE DONNEES envoie 10 dans les deux bits de poids le plus faible du mot d'état de la trame pour montrer qu'il a été sélectionné et qu'il a survécu aux arbitrages de bus série. Finalement, l'EMETTEUR DE DONNEES surveille le bit de détection de désynchronisation. Si le bit est 0, l'EMETTEUR DE DONNEES active sa sortie DSENT pour indiquer qu'il a envoyé ses données avec succès.

6.1.4 Parleur transactionnel

Comme le montre la figure 6-4, page 188, un parleur transactionnel est semblable au parleur sur demande décrit au paragraphe 6.1.3. Il diffère en ce que l'entrée ENABLE S du RECEPTEUR D'EN-TETE n'est pas en permanence à l'état vrai, mais est plutôt commandée par une bascule qui indique si les couches de niveau élevé du gestionnaire du bus série ont chargé les données à transmettre dans une trame courante du bus série.

The DATA SENDER then sends one of the values 001-110 in the Frame Type subframe, according to how many bytes were loaded into the Data Port by higher layers of serial bus management. The DATA SENDER uses serial bus arbitration logic while sending the Frame Type. If it loses the arbitration to the value 111, indicating a Cancelled frame, the DATA SENDER and HEADER RECEIVER both ignore the frame.

If the DATA SENDER loses the serial bus arbitration in the Frame Type field to a larger value 010-110, it waits for the number of bytes indicated by the winning value, and then sends 110 in the Frame Status field to signal a "Sender/Sender Data Size Conflict".

Typically the DATA SENDER wins the arbitration in the Frame Type field and then sends the indicated number of bytes immediately thereafter. It may or may not use serial bus arbitration logic while sending the data, depending on the DSAE bit in the Header which is reflected in the DSAE signal from the HEADER RECEIVER to the DATA SENDER.

If the DATA SENDER uses serial bus arbitration logic and loses the arbitration in the data field, it ignores the rest of the frame. Otherwise, after the data it monitors the following MSB of the Frame Status. If the MSB is 1, the DATA SENDER and HEADER RECEIVER ignore the rest of the frame. If the MSB is 0, the DATA SENDER sends 10 in the least significant two bits of the Frame Status, to show that it was selected and that it survived any serial bus arbitration. Finally, the DATA SENDER monitors the Jam Detect bit. If the bit is 0, the DATA SENDER activates its DSENT output to indicate that it has sent data successfully.

6.1.4 Transaction Talker

As shown in Figure 6-4, page 189, a Transaction Talker is similar to the On-Demand Talker described in Paragraph 6.1.3. It differs in that the HEADER RECEIVER's ENABLE S input is not permanently set True, but instead is controlled by a latch reflecting whether higher layers of serial bus management have loaded data to be transmitted in a current serial bus frame.

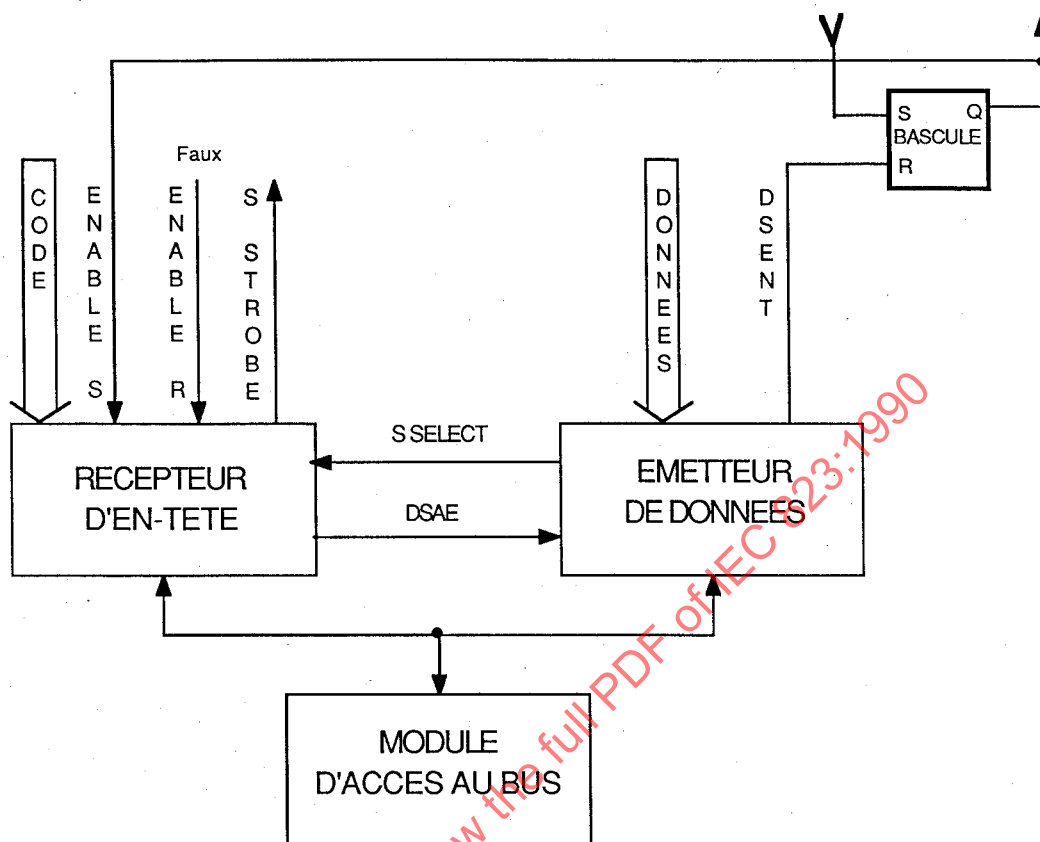


Figure 6-4 - Parleur transactionnel

6.1.4.1 But

Un parleur transactionnel peut être utilisé pour envoyer des données aux écouteurs sur d'autres cartes, dans des applications où les données de chaque trame doivent être transmises avec succès avant que d'autres données des couches supérieures du gestionnaire du bus série puissent être acceptées.

6.1.4.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un parleur transactionnel de la façon suivante:

- 1) Elles positionnent le port de code du RECEPTEUR D'EN-TETE à une valeur comprise entre 1 et 1022.
- 2) Elles mettent à zéro la bascule connectée à ENABLE S, si bien que ENABLE S est à l'état faux.
- 3) Elles mettent ENABLE R à l'état faux.

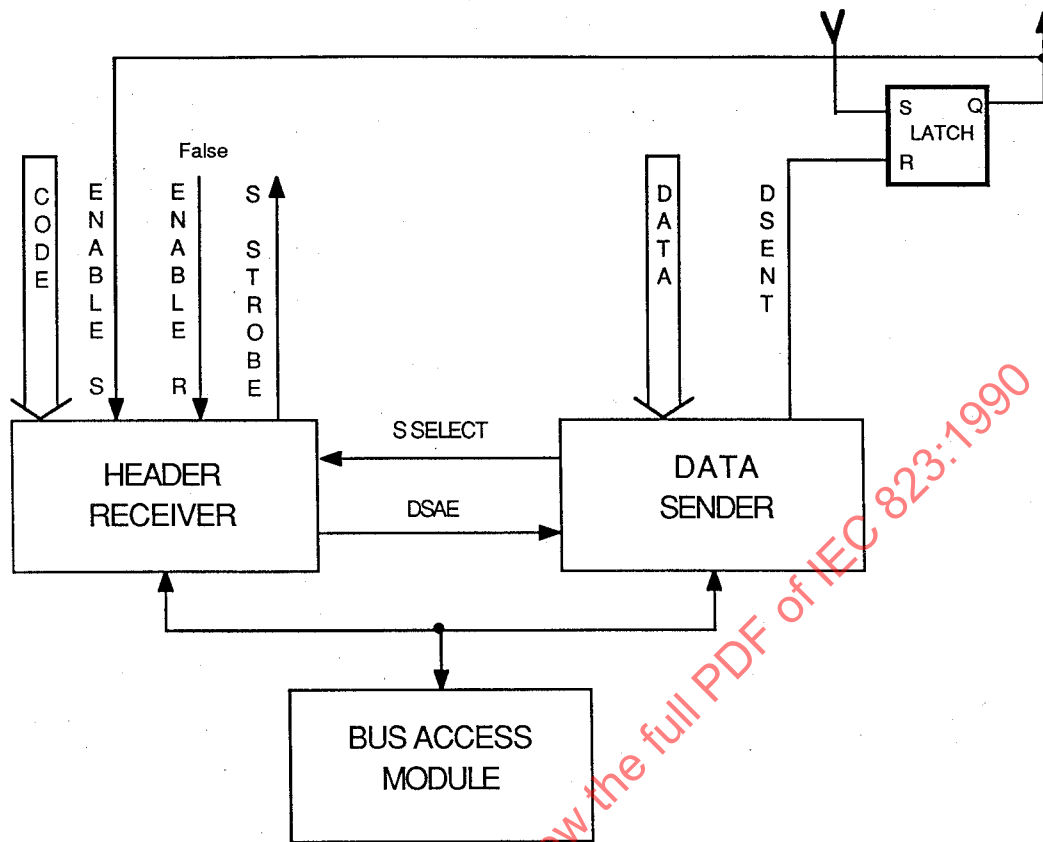


Figure 6-4 - Transaction Talker

6.1.4.1 Purpose

A Transaction Talker can be used to send data to Listeners on other boards in applications where the data for each frame have to be transmitted successfully before further data can be accepted from higher layers of serial bus management.

6.1.4.2 Operation

Higher layers of serial bus management configure a Transaction Talker as follows:

- 1) They set the Code Port of the HEADER RECEIVER to some value in the range 1-1022.
- 2) They reset the latch connected to ENABLE S, so that ENABLE S is False.
- 3) They make ENABLE R False.

Ensuite, chaque fois que les couches supérieures du gestionnaire du bus série chargent des données dans le port de données de l'EMETTEUR DE DONNEES, elles positionnent la bascule, si bien que ENABLE S devient vrai. Les couches supérieures attendent alors pour charger d'autres données jusqu'à ce que l'EMETTEUR DE DONNEES ait positionné DSENT, ce qui remet la bascule à zéro.

Si un groupe gestionnaire, quelque part sur le bus série, envoie un en-tête avec l'adresse du groupe dans le champ S, le RECEPTEUR D'EN-TETE reconnaît l'adresse. L'action du parleur transactionnel dépend alors de l'état positionné de la bascule externe, c'est-à-dire du fait que les couches supérieures du gestionnaire du bus série ont chargé des données pour cette trame dans l'EMETTEUR DE DONNEES.

Sinon, ENABLE S est faux, et le RECEPTEUR D'EN-TETE annule la trame en envoyant 111 dans le champ de type de trame. Cela termine la trame.

S'il y a des données à envoyer, ENABLE S est vrai et le RECEPTEUR D'EN-TETE n'annule pas la trame, mais positionne plutôt S SELECT pour l'EMETTEUR DE DONNEES.

L'EMETTEUR DE DONNEES envoie alors une des valeurs 001-110 dans la sous-trame de type de trame selon le nombre d'octets qui ont été chargés dans le port de donnée par les couches supérieures du gestionnaire du bus série. L'EMETTEUR DE DONNEES utilise la logique d'arbitrage du bus série pendant qu'il envoie le type de trame. S'il perd l'arbitrage à la valeur 111, indiquant une trame annulée, l'EMETTEUR DE DONNEES et le RECEPTEUR D'EN-TETE ignorent tous deux la trame.

Si l'EMETTEUR DE DONNEES perd l'arbitrage du bus série dans le champ de type de trame à une valeur plus grande que celles de l'intervalle 010-110, il attend le nombre d'octets indiqué par la valeur gagnante et puis envoie 110 dans le champ de type de trame pour signaler un "conflit de dimension de données émetteur/émetteur".

Typiquement, l'EMETTEUR DE DONNEES gagne l'arbitrage dans le champ de type de trame et puis envoie le nombre d'octets indiqué immédiatement après. Il peut ou non utiliser la logique d'arbitrage du bus série pendant qu'il envoie les données, selon l'état du bit DSAE dans l'en-tête qui est répété dans le signal DSAE du RECEPTEUR D'EN-TETE vers l'EMETTEUR DE DONNEES.

Si l'EMETTEUR DE DONNEES utilise la logique d'arbitrage du bus série et perd l'arbitrage dans le champ de données, il ignore le reste de la trame. Autrement, après les données, il surveille le bit le plus significatif suivant du mot d'état de la trame. Si le MSB est à 1, l'EMETTEUR DE DONNEES et le RECEPTEUR D'EN-TETE ignorent le reste de la trame. Si le MSB est à 0, l'EMETTEUR DE DONNEES envoie 10 dans les deux bits de poids le plus faible du mot d'état de la trame pour montrer qu'il a été sélectionné et qu'il a survécu aux arbitrages du bus série. Finalement, l'EMETTEUR DE DONNEES surveille le bit de détection de désynchronisation. Si le bit est à 0 l'EMETTEUR DE DONNEES active sa sortie DSENT, ce qui remet à zéro la bascule externe.

Thereafter, each time higher layers of serial bus management load data into the DATA SENDER's Data Port, they then set the latch, so that ENABLE S goes True. The higher layers then wait to load further data until the DATA SENDER asserts DSENT, resetting the latch.

If a Controller group anywhere on the serial bus sends a Header with this group's address in the S field, the HEADER RECEIVER recognizes the address. The action of a Transaction Talker now depends on whether the external latch is set, i.e. on whether higher layers of serial bus management have loaded data for this frame into the DATA SENDER.

If not, ENABLE S is False, and the HEADER RECEIVER cancels the frame by sending 111 in the Frame Type field. This completes the frame.

If there is data to be sent, ENABLE S is True and the HEADER RECEIVER does not cancel the frame, but instead asserts S SELECT to the DATA SENDER.

The DATA SENDER then sends one of the values 001-110 in the Frame Type subframe, according to how many bytes were loaded into the Data Port by higher layers of serial bus management. The DATA SENDER uses serial bus arbitration logic while sending the Frame Type. If it loses the arbitration to the value 111, indicating a Cancelled frame, the DATA SENDER and HEADER RECEIVER both ignore the frame.

If the DATA SENDER loses the serial bus arbitration in the Frame Type field to a larger value 010-110, it waits for the number of bytes indicated by the winning value, and then sends 110 in the Frame Status field to signal a "Sender/Sender Data Size Conflict".

Typically the DATA SENDER wins the arbitration in the Frame Type field and then sends the indicated number of bytes immediately thereafter. It may or may not use serial bus arbitration logic while sending the data, depending on the DSAE bit in the Header which is reflected in the DSAE signal from the HEADER RECEIVER to the DATA SENDER.

If the DATA SENDER uses serial bus arbitration logic and loses the arbitration in the data field, it ignores the rest of the frame. Otherwise, after the data it monitors the following MSB of the Frame Status. If the MSB is 1, the DATA SENDER and HEADER RECEIVER ignore the rest of the frame. If the MSB is 0, the DATA SENDER sends 10 in the least significant two bits of the Frame Status, to show that it was selected and that it survived any serial bus arbitration. Finally, the DATA SENDER monitors the Jam Detect bit. If the bit is 0, the DATA SENDER activates its DSENT output, which clears the external latch.

La remise à zéro de la bascule a deux effets: signaler aux couches supérieures du gestionnaire du bus série que les données chargées précédemment ont été envoyées et qu'il peut charger d'autres données si besoin est. Deuxièmement, mettre l'entrée ENABLE S à l'état faux, si bien que, si une autre trame sélectionne ce parleur transactionnel avant que les couches supérieures aient chargé de nouvelles données, le groupe annulera la trame.

6.1.5 Ecouteur sur demande

Comme le montre la figure 6-5 ci-dessous, un écouteur sur demande est composé d'un RECEPTEUR D'EN-TETE et d'un RECEPTEUR DE DONNEES avec l'entrée ENABLE R du RECEPTEUR D'EN-TETE positionné à l'état vrai. Cela montre que, dès que le système de bus série est initialisé et mis en fonctionnement, le groupe est toujours validé pour recevoir des données en réponse à un en-tête qui le sélectionne dans l'adresse R.

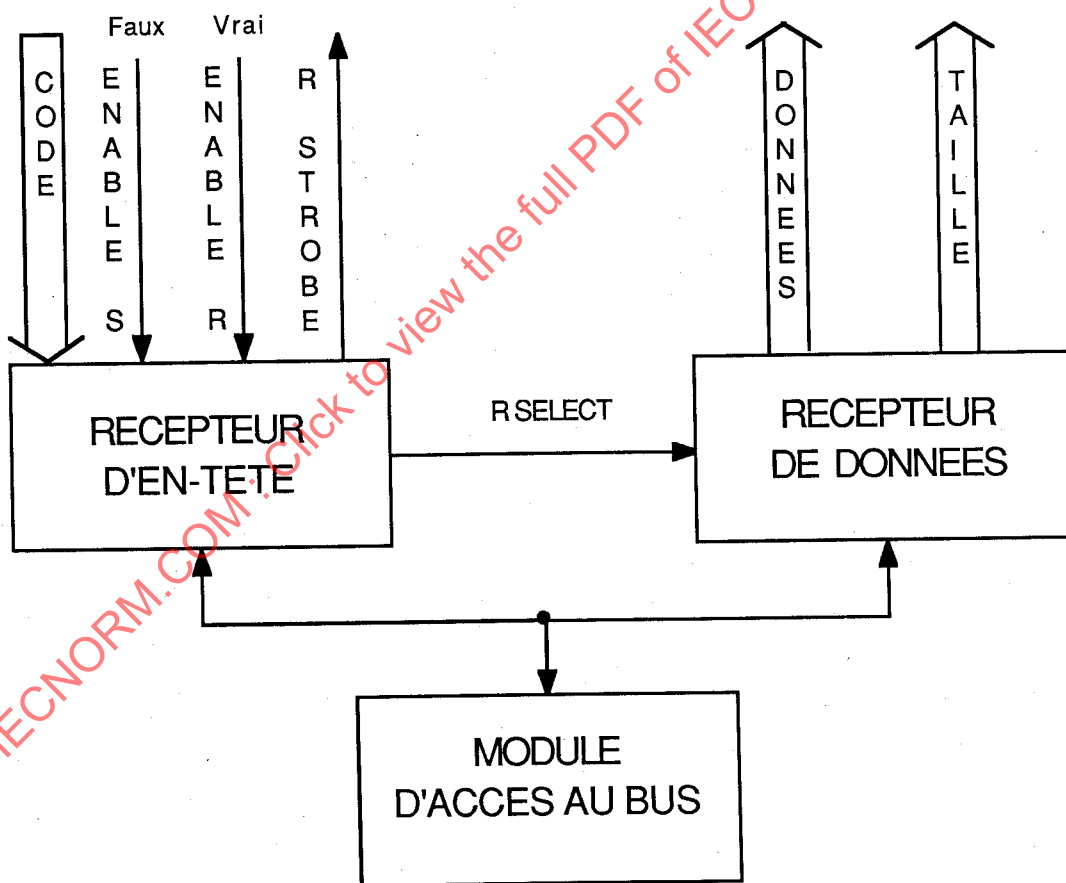


Figure 6-5 - Ecouteur sur demande

Clearing the latch has two effects: first, it signals higher layers of serial bus management that the previously loaded data has been sent and that it can load further data if appropriate. Second, it makes the ENABLE S input False so that, if another frame selects this Transaction Talker before higher layers load new data, the group will cancel the frame.

6.1.5 On-Demand Listener

As shown in Figure 6-5, an On-Demand Listener is composed of a HEADER RECEIVER and a DATA RECEIVER, with the ENABLE R input of the HEADER RECEIVER set to True. This indicates that, once the serial bus system is initialized and set into operation, the group is always enabled to receive data in response to a Header that selects it in the R address.

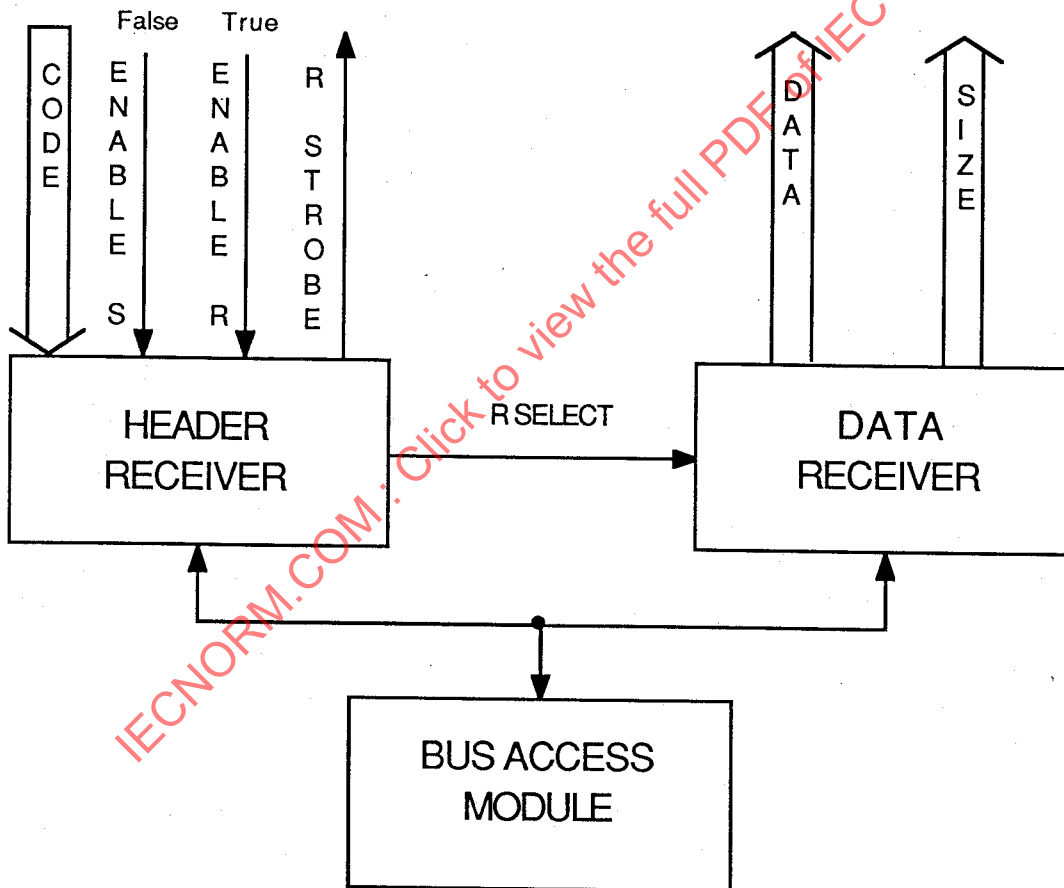


Figure 6-5 - On-Demand Listener

6.1.5.1 But

Un écouteur sur demande peut être utilisé pour prendre des données venant de parleurs sur d'autres cartes, dans des applications où la donnée est simplement utilisée par les couches supérieures du gestionnaire du bus série "chaque fois qu'on a besoin de la valeur courante de la donnée", sans tenir compte du moment où la donnée a été envoyée. Ce procédé est avantageux quand on a souvent besoin des données, mais qu'elles changent rarement.

Plusieurs écouteurs sur demande peuvent être utilisés avec un gestionnaire d'écriture (voir paragraphe 6.2.1) pour diffuser la valeur courante d'un paramètre critique du système à des processus parallèles commandés par les cartes de l'écouteur, sur l'ordre d'un processeur sur la carte avec le gestionnaire.

6.1.5.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un écouteur sur demande de la façon suivante:

- 1) Elles positionnent le port de code du RECEPTEUR D'EN-TETE à une valeur comprise entre 1 et 1022.
- 2) Elles mettent ENABLE S à l'état faux.
- 3) Elles mettent ENABLE R à l'état vrai.

De plus périodiquement, les couches supérieures du gestionnaire du bus série lisent et utilisent les données présentées au port de données du RECEPTEUR DE DONNEES. Cette utilisation n'a aucun effet sur l'état de l'écouteur sur demande. Avec ce groupe, il n'y a pas de relation définie entre le moment où la donnée est lue et celui où elle est reçue sur le bus série.

REGLE 6.4:

Quand les couches supérieures du gestionnaire du bus série lisent des données venant d'un écouteur sur demande, elles NE DOIVENT PAS pouvoir lire *accidentellement* un mélange de données d'une nouvelle et d'une ancienne trame.

SUGGESTION 6.2:

Concevoir les écouteurs sur demande avec un bit d'état que les couches supérieures mettent à zéro avant de commencer à lire les données, et qui est mis à 1 quand la nouvelle donnée a été reçue. Après avoir terminé la lecture des données, les couches supérieures peuvent alors vérifier le bit d'état et ignorer les données lues si le bit est positionné à 1.

Si un groupe gestionnaire, quelque part sur le bus série, envoie un en-tête avec l'adresse d'un écouteur sur demande dans le champ R, le RECEPTEUR D'EN-TETE reconnaît l'adresse. Puisque ENABLE R est toujours vrai, le RECEPTEUR D'EN-TETE n'annule pas la trame mais positionne R SELECT pour le RECEPTEUR DE DONNEES.

6.1.5.1 Purpose

An On-Demand Listener can be used to capture data from Talkers on other boards in applications where the data is simply used by higher layers of serial bus management "whenever the current value of the data is needed", without regard for when the data was sent. Thus, this scheme is advantageous when data is needed often but seldom changes.

Several On-Demand Listeners might be used with a Writing Controller (see Paragraph 6.2.1), to broadcast the current value of a critical system parameter to parallel processes being controlled by the Listeners' boards, on command from a processor on the board with the Controller.

6.1.5.2 Operation

Higher layers of serial bus management configure an On-Demand Listener as follows:

- 1) They set the Code Port of the HEADER RECEIVER to some value in the range 1-1022.
- 2) They make ENABLE S False.
- 3) They make ENABLE R True.

Periodically, higher layers of serial bus management also read out and use the data presented at the DATA RECEIVER's Data Port. This use has no effect on the state of the On-Demand Listener. With this group, there is no defined relationship between when data is read out and when it is received on the serial bus.

RULE 6.4:

When higher layers of serial bus management are reading data out of an On-Demand Listener, they MUST NOT be allowed to *inadvertently* read out a mixture of data from an old frame and a new frame.

SUGGESTION 6.2:

Design On-Demand Listeners with a status bit that higher layers clear before they begin to read out data, and which is set when new data is received. After they complete reading out data, the higher layers can then check the status bit and ignore the data they have read out if the bit is set.

If a Controller group anywhere on the serial bus sends a Header with the On-Demand Listener's address in the R field, the HEADER RECEIVER recognizes the address. Since ENABLE R is always True, the HEADER RECEIVER does not cancel the frame, but instead asserts R SELECT to the DATA RECEIVER.

R SELECT fait que le RECEPTEUR DE DONNEES prend le code de type de trame. Si le code de type de trame est 111, ce qui indique que d'autres groupes sélectionnés ont annulé la trame, le RECEPTEUR D'EN-TETE et le RECEPTEUR DE DONNEES ignorent tous deux la trame.

Si le code de type de trame est 000, ce qui indique qu'aucun groupe parleur n'est sélectionné, le RECEPTEUR DE DONNEES envoie 101 dans le champ du mot d'état de trame suivant pour signaler le problème. Si le code de type de trame indique plus de données que le RECEPTEUR DE DONNEES ne peut en traiter, il attend que le nombre d'octets indiqué soit passé, puis envoie 101 dans le code du mot d'état de la trame.

Si le code du mot d'état de la trame indique une trame de transfert de données d'une longueur telle que le RECEPTEUR DE DONNEES peut le traiter, celui-ci prend les octets de données qui suivent. Puis il teste les deux premiers bits du champ du mot d'état de trame. Si le bit le plus significatif est à zéro, il envoie alors un 1 dans le dernier bit du mot d'état de la trame pour indiquer sa présence. Il prend ensuite le bit de détection de désynchronisation. Si le bit de détection de désynchronisation est 0 et que les deux premiers bits du mot d'état de la trame sont 01, la trame est transmise avec succès. Dans ce cas, le parleur sur demande rend les données enregistrées disponibles à son port de données pour l'utilisation par les couches supérieures du gestionnaire du bus série. La sortie R STROBE du RECEPTEUR D'EN-TETE indique l'arrivée des données.

6.1.6 *Ecouteur transactionnel*

Comme le montre la figure 6-6, page 198, un écouteur transactionnel est semblable à l'écouteur sur demande décrit dans le paragraphe précédent. Il en diffère en ce que l'entrée ENABLE R du RECEPTEUR D'EN-TETE n'est pas positionnée en permanence à l'état vrai, mais est plutôt commandée par une bascule indiquant si les couches supérieures du gestionnaire du bus série ont traité ou transmis les données précédemment reçues.

6.1.6.1 *But*

Un écouteur transactionnel peut être utilisé pour prendre des données venant de parleurs sur d'autres cartes, dans des applications où les données ont été traitées par les couches supérieures du gestionnaire du bus série de la carte de l'écouteur avant que de nouvelles données puissent être acceptées.

R SELECT makes the DATA RECEIVER capture the Frame Type code. If the Frame Type code is 111, indicating that some other selected group cancelled the frame, both the HEADER RECEIVER and DATA RECEIVER ignore the frame.

If the Frame Type code is 000, indicating that no Talker group is selected, the DATA RECEIVER sends 101 in the following Frame Status field to signal the problem. If the Frame Type code indicates more data than the DATA RECEIVER is designed to handle, it waits for the indicated number of bytes to pass and then sends 101 in the Frame Status code.

If the Frame Status code indicates a Data Transfer frame of a length that the DATA RECEIVER can handle, it captures the following data bytes. It then monitors the first two bits of the Frame Status field. If the MSB is zero, it then sends a 1 in the last bit of the Frame Status to indicate its presence. It then captures the Jam Detect bit. If the Jam Detect bit is 0 and the first two bits of the Frame Status were 01, the frame is successful. In this case, the On-Demand Talker makes the stored data available at its Data Port for the use of higher layers of serial bus management. The R STROBE output of the HEADER RECEIVER indicates the arrival of the data.

6.1.6 *Transaction Listener*

As shown in Figure 6-6, page 199, a Transaction Listener is similar to the On-Demand Listener described in the previous paragraph. It differs in that the HEADER RECEIVER's ENABLE R input is not permanently set True, but instead is controlled by a latch reflecting whether higher layers of serial bus management have processed or handled previously-received data.

6.1.6.1 *Purpose*

A Transaction Listener can be used to capture data from Talkers on other boards, in applications where the data has to be processed by higher layers of serial bus management on the Listener's board before new data can be accepted.

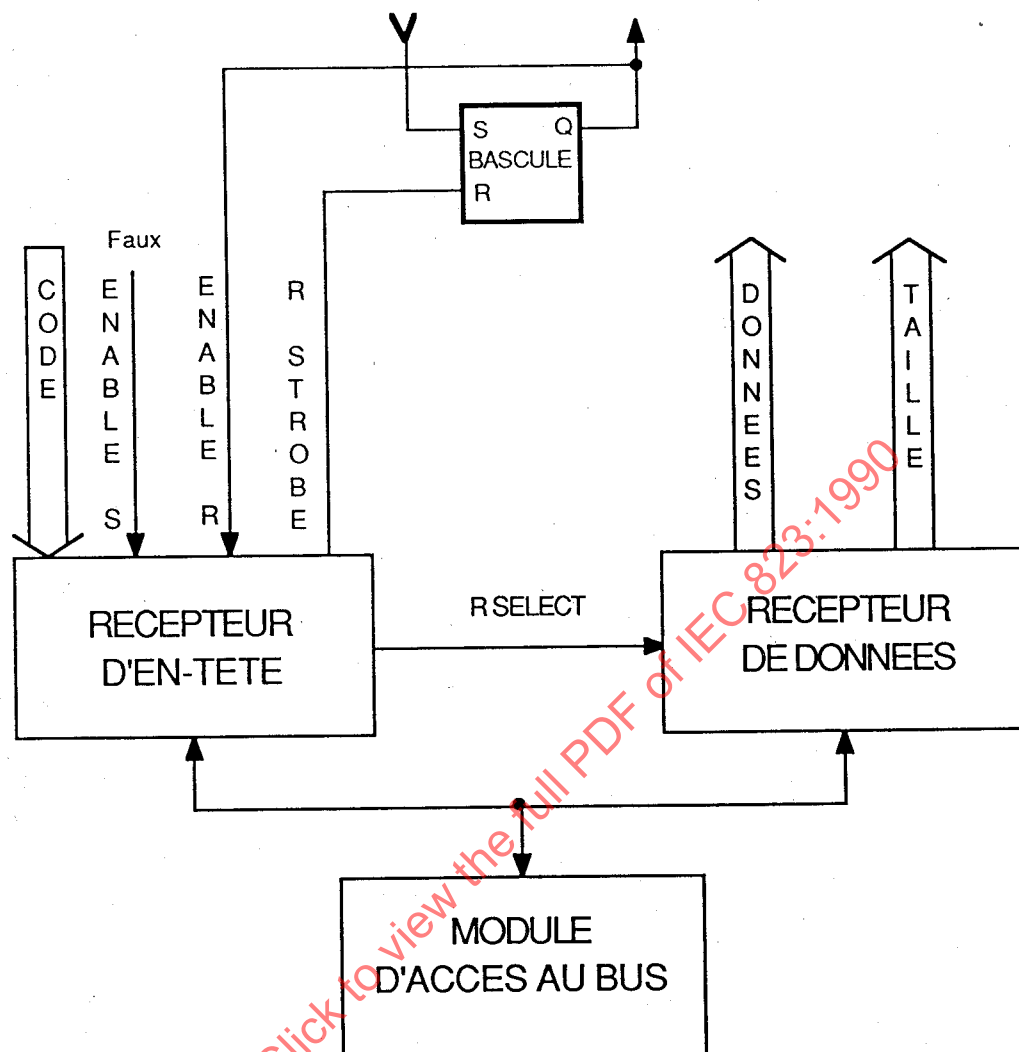


Figure 6-6 - Ecouteur transactionnel

6.1.6.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un écouteur transactionnel de la façon suivante:

- 1) Elles positionnent le port de code du RECEPTEUR D'EN-TETE à une valeur comprise entre 1 et 1022.
- 2) Elles mettent ENABLE S à l'état faux.
- 3) Elles positionnent la bascule reliée à ENABLE R de sorte que ENABLE R soit vrai.

Si un groupe gestionnaire, quelque part sur le bus série, envoie un en-tête avec l'adresse de l'écouteur transactionnel dans le champ R, le RECEPTEUR D'EN-TETE reconnaît l'adresse. L'action de l'écouteur transactionnel dépend du fait que la bascule externe est positionnée à un, c'est-à-dire si les couches supérieures du gestionnaire du bus série ont traité les données qu'elles ont reçues dans la dernière trame de ce type.

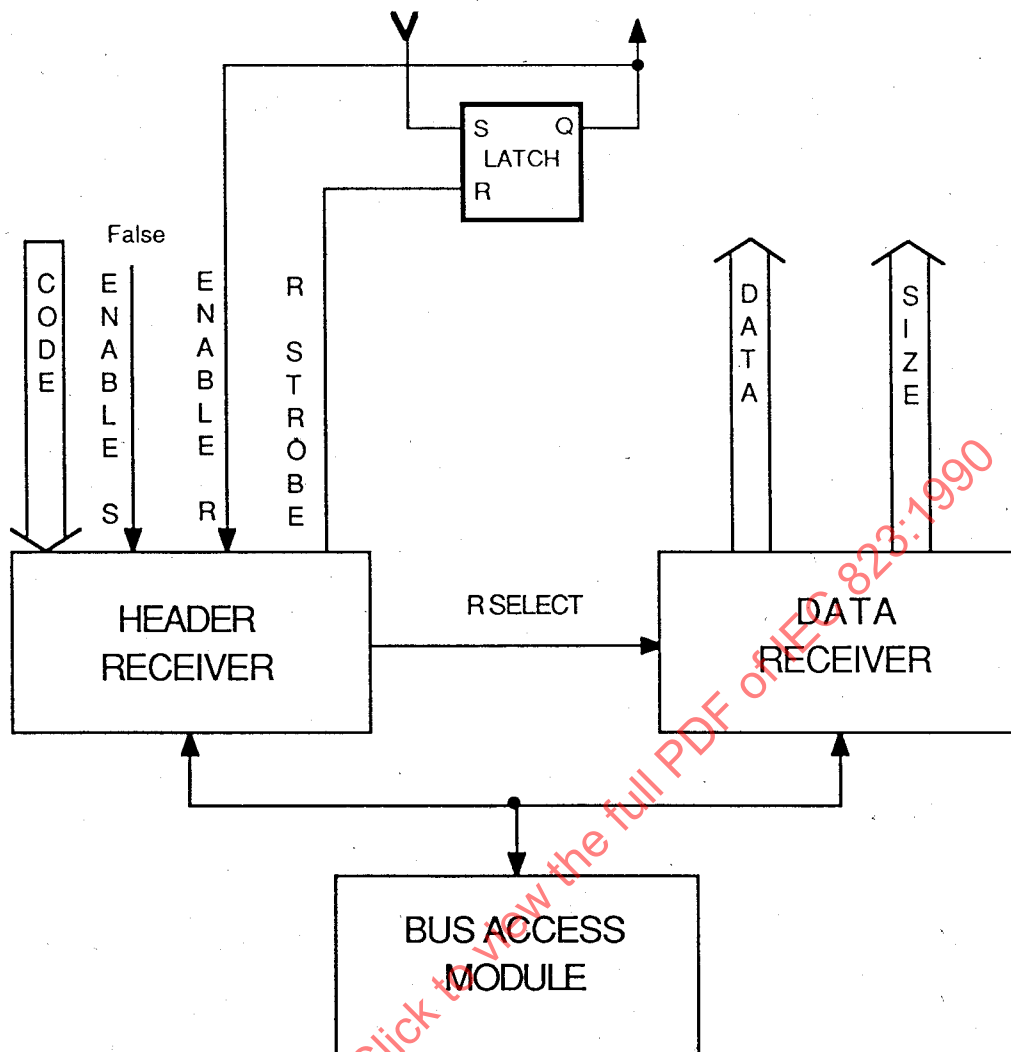


Figure 6-6 - Transaction Listener

6.1.6.2 Operation

Higher layers of serial bus management configure a Transaction Listener as follows:

- 1) They set the Code Port of the HEADER RECEIVER to some value in the range 1-1022.
- 2) They make ENABLE S False.
- 3) They set the latch connected to ENABLE R so that ENABLE R is True.

If a Controller group anywhere on the serial bus sends a Header with the Transaction Listener's address in the R field, the HEADER RECEIVER recognizes the address. The action of a Transaction Listener now depends on whether the external latch is set, i.e. on whether higher layers of serial bus management have processed the data that was received in the last frame of this type.

Si elle ne l'est pas, ENABLE R est faux et le RECEPTEUR D'EN-TETE annule la trame en envoyant 111 dans le champ de type de trame. Cela termine la trame.

Si elle l'est, ENABLE R est vrai et le RECEPTEUR D'EN-TETE n'annule pas la trame mais positionne R SELECT du RECEPTEUR DE DONNEES.

Le RECEPTEUR DE DONNEES surveille alors le code dans le champ de type de trame et le fonctionnement se poursuit comme précédemment pour l'écouteur sur demande. Si la transmission de la trame est bonne, le RECEPTEUR D'EN-TETE active sa sortie R STROBE qui remet à zéro la bascule externe. Cela a deux effets: premièrement, il signale aux couches supérieures du gestionnaire du bus série que les données sont arrivées et qu'il convient de les traiter. Deuxièmement, il met l'entrée ENABLE R à l'état faux de sorte que, si une autre trame sélectionne ce parleur transactionnel avant que les couches supérieures aient traité les données, le groupe annulera la trame.

Quand les couches supérieures ont lu les données issues du RECEPTEUR DE DONNEES et les ont traitées, elles positionnent à un la bascule, ce qui met l'entrée ENABLE R à l'état vrai et valide la réception de nouvelles données.

6.1.7 Indicateur multiadresse

Un indicateur multiadresse est représenté par la figure 6-7 ci-dessous. Il est semblable à l'indicateur simple décrit au paragraphe 6.1.1, excepté que les sorties S STROBE et R STROBE de plusieurs RECEPTEURS D'EN-TETE sont câblés en OU logique pour fournir les entrées de mise à un et de mise à zéro de la bascule.

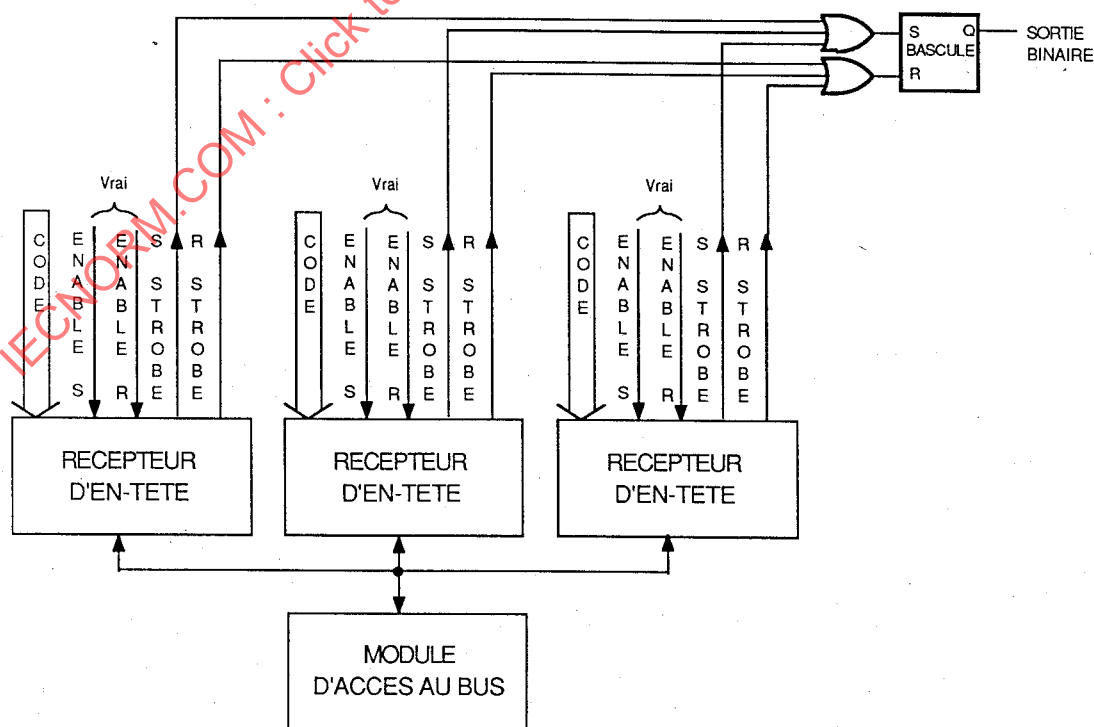


Figure 6-7 - Indicateur multiadresse

If not, ENABLE R is False and the HEADER RECEIVER cancels the frame by sending 111 in the Frame Type field. This completes the frame.

If so, ENABLE R is True and the HEADER RECEIVER does not cancel the frame, but instead asserts R SELECT to the DATA RECEIVER.

The DATA RECEIVER then monitors the code in the Frame Type field, and operation proceeds as described previously for the On-Demand Listener. If the frame transmission is successful, the HEADER RECEIVER activates its R STROBE output, which clears the external latch. This has two effects: first, it signals higher layers of serial bus management that data has arrived and should be processed. Second, it makes the ENABLE R input False so that, if another frame selects this Transaction Talker before higher layers process the data, the group will cancel the frame.

When higher layers have read the data out of the DATA RECEIVER and processed it, they set the latch, making the ENABLE R input True and enabling the reception of new data.

6.1.7 Multiaddress Flag

A Multiaddress Flag is shown in Figure 6-7. It is similar to the Simple Flag described in Paragraph 6.1.1 except that the S STROBE and R STROBE outputs of several HEADER RECEIVERS are each logically OR'ed to produce the set and reset inputs of the latch.

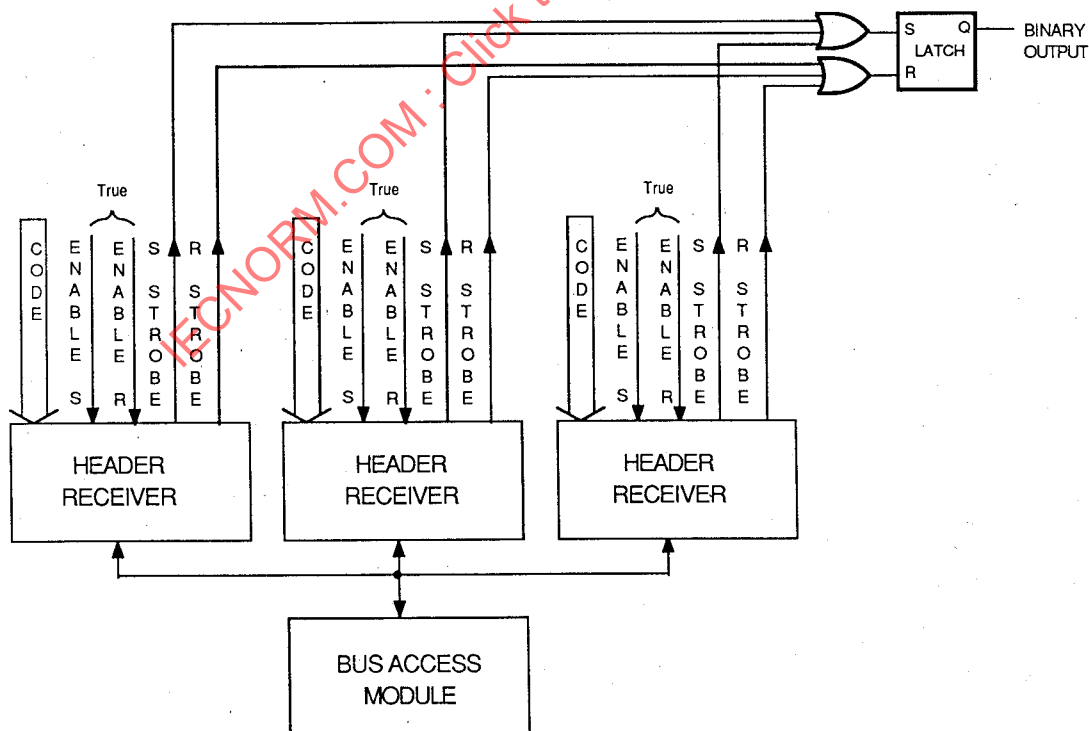


Figure 6-7 - Multiaddress Flag

6.1.7.1 But

Un indicateur multiadresse peut être utilisé partout où cela est nécessaire pour un signal de sortie ou un bit de registre qui puisse être adressé selon différentes possibilités. Par exemple, dans un système tolérant aux fautes, un indicateur multiadresse pourrait être utilisé pour valider/invalider l'interface bus système d'une carte, donnant aux couches supérieures du gestionnaire du bus série la possibilité d'invalider de façon dynamique "tous les maîtres", une carte particulière, ou "toutes les cartes produites par un fabricant XYZ".

6.1.7.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un indicateur multiadresse comme suit:

- 1) Elles positionnent le port de code de chaque RECEPTEUR D'EN-TETE à une valeur différente comprise entre 1 et 1022.
- 2) Elles mettent les entrées ENABLE S et ENABLE R de chaque RECEPTEUR D'EN-TETE à l'état vrai.

SUGGESTION 6.3:

Pour réaliser l'affectation des adresses dans un système de bus série avec différents niveaux de groupe d'adressage, le faire de façon qu'elle soit en rapport avec les adresses d'un groupe. S'il y a un maximum de "n" groupes de modules de bus série dans le groupe de niveau immédiatement supérieur, faire l'adresse de groupe sous la forme binaire XXXXX00000, avec suffisamment de poids faibles à zéro afin que les groupes de modules puissent être numérotés XXXXX00001, XXXXX00010, ..., XXXXX "n". Utiliser la même procédure (avec davantage de poids faibles à zéro) pour former les adresses de groupe de niveau supérieur.

Quand un groupe gestionnaire, quelque part sur le bus série, envoie un en-tête dans lequel l'adresse S (ou R) sélectionne l'un des RECEPTEUR D'EN-TETE, ce RECEPTEUR D'EN-TETE surveille le type de trame ainsi que l'état de la trame, comme décrit pour un indicateur simple (paragraphe 6.1.1). Si la trame est transmise correctement, le RECEPTEUR D'EN-TETE place sa sortie S STROBE (ou R STROBE) à l'état vrai, positionnant à un (ou à zéro) la bascule.

6.1.8 Parleur multiadresse

Un parleur multiadresse est représenté par la figure 6-8, page 204. Il diffère des types de parleurs décrits précédemment par le fait que les sorties S SELECT de plusieurs RECEPTEURS D'EN-TETE sont câblés en OU logique pour fournir l'entrée S SELECT de l'EMETTEUR DE DONNEES.

OBSERVATION 6.1:

La propriété multiadresse est indépendante des autres caractéristiques d'un parleur, écouteur ou indicateur. Ainsi, il peut exister des parleurs multiadresse sur demande et des parleurs multiadresse transactionnels, etc.

6.1.7.1 Purpose

A Multiaddress Flag can be used wherever there is a need for an output signal or register bit that can be addressed in any of several ways. For example, in a fault-tolerant system a Multiaddress Flag could be used to enable/disable the system bus interface of a board, giving higher layers of serial bus management the ability to dynamically disable "all Masters", a particular board or "all boards made by manufacturer XYZ".

6.1.7.2 Operation

Higher layers of serial bus management configure a Multiaddress Flag as follows:

- 1) They set the Code Port of each HEADER RECEIVER to a different value in the range 1-1022.
- 2) They make the ENABLE S and ENABLE R inputs of each HEADER RECEIVER True.

SUGGESTION 6.3:

When assigning addresses in a serial bus system with various levels of group addressing, do so in a way that relates the addresses in a group. If there are a maximum of "n" serial bus module groups in the next-higher grouping, make the group address of the binary form XXXXX00000, with enough low order zeros so that the module groups can be numbered XXXXX00001, XXXXX00010, ..., XXXXX "n". Use the same procedure (with more low order zeros) to form still higher group addresses.

Whenever a Controller group anywhere on the serial bus sends a Header in which the S (or R) address selects one of the HEADER RECEIVERS, that HEADER RECEIVER monitors the Frame Type and typically also the Frame Status, as described for a Simple Flag in Paragraph 6.1.1. If the frame is successful, the HEADER RECEIVER makes its S STROBE (or R STROBE) output True, setting (or resetting) the latch.

6.1.8 Multiaddress Talker

A Multiaddress Talker is shown in Figure 6-8, page 205. It differs from the types of Talkers described previously in that the S SELECT outputs of several HEADER RECEIVERS are logically OR'ed to produce the S SELECT input of the DATA SENDER.

OBSERVATION 6.1:

The Multiaddress property is independent of other features of a Talker, Listener or Flag. Thus, there can be Multiaddress On-Demand Talkers and Multiaddress Transaction Talkers, etc.

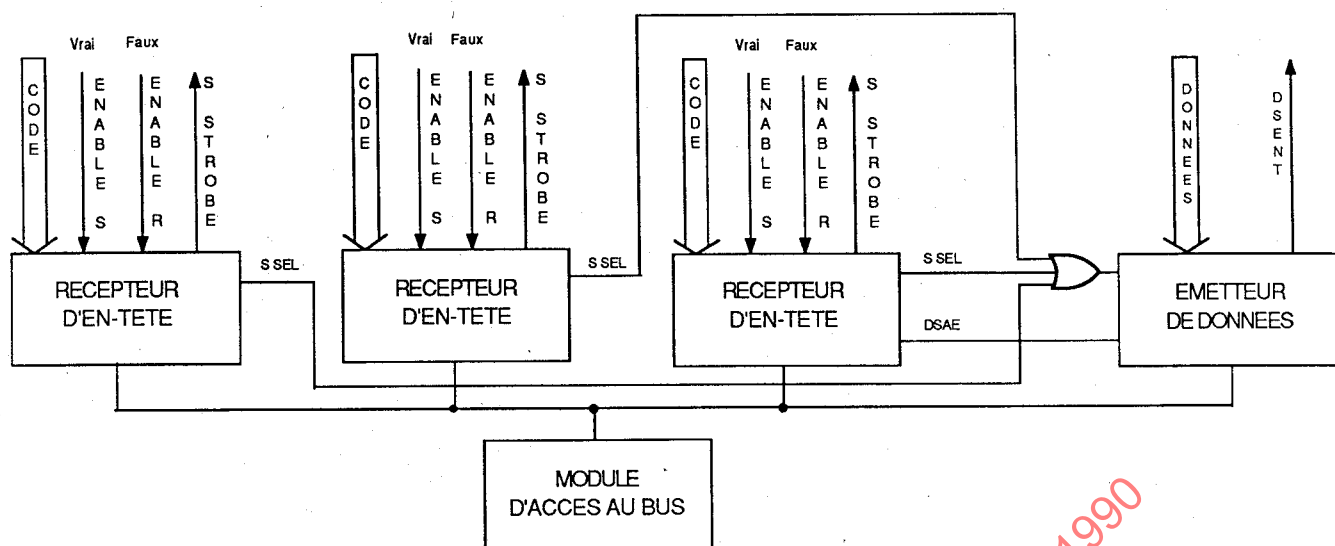


Figure 6-8 - Parleur multiadresse sur demande

6.1.8.1 But

Un parleur multiadresse peut être utilisé dans des applications où les données doivent être émises d'une source qui peut être adressée de plusieurs manières. Par exemple, dans un système de surveillance, un capteur de température peut être lisible directement en utilisant sa propre adresse unique, ou la température la plus élevée parmi plusieurs capteurs peut être déterminée par la lecture de "tous les capteurs" ou de "tous les capteurs en zone 6".

6.1.8.2 Fonctionnement

Les couches supérieures du gestionnaire du bus série configurent un parleur multiadresse comme suit:

- 1) Elles positionnent les ports de code des RECEPTEURS D'EN-TETE à différentes valeurs comprises entre 1 et 1022.
- 2) Elles mettent l'entrée ENABLE R de chaque RECEPTEUR D'EN-TETE à l'état faux.
- 3) Pour un parleur multiadresse sur demande, elles chargent la donnée dans le port de donnée de l'EMETTEUR DE DONNEES et puis mettent ENABLE S à l'état vrai. Pour un parleur multiadresse transactionnel, elles mettent à zéro la bascule commandant ENABLE S de manière que ENABLE S soit à l'état faux.

Le fonctionnement d'un parleur multiadresse est semblable à celui décrit au paragraphe 6.1.4 pour le type correspondant de parleur (adresse simple), excepté que l'EMETTEUR DE DONNEES est sélectionné quand un quelconque des RECEPTEURS D'EN-TETE reconnaît son adresse dans le champ S d'un en-tête. Dans la figure 6-8 ci-dessus, l'entrée S SELECT de l'EMETTEUR DE DONNEES est à l'état vrai quand l'une des sortie S SELECT des RECEPTEURS D'EN-TETE est à l'état vrai. Pour un parleur multiadresse transactionnel, il y a une simple bascule signalant si une donnée a été fournie par les couches supérieures du gestionnaire du bus série - sa sortie est connectée aux entrées ENABLE S de tous les RECEPTEURS D'EN-TETE.

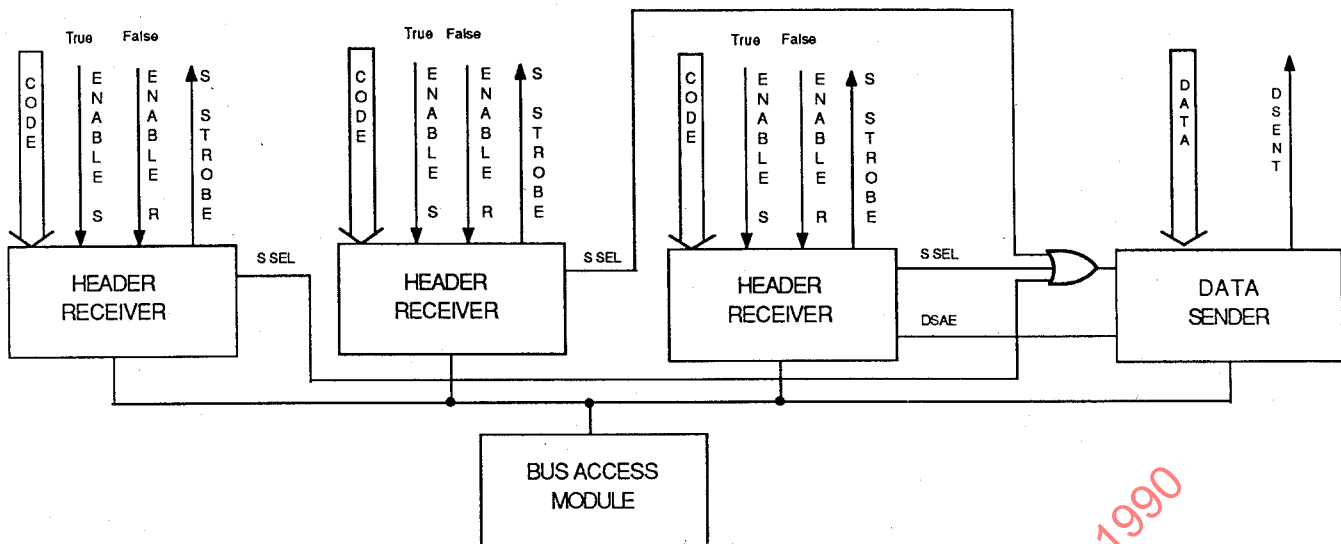


Figure 6-8 - Multiaddress On-Demand Talker

6.1.8.1 Purpose

A Multiaddress Talker can be used in applications where data is to be sent from a source that can be addressed in several ways. For example, in a monitoring system, a temperature sensor might be readable directly by using its own unique address, or the largest temperature among several sensors might be determined by reading from "all sensors" or "all sensors in zone 6".

6.1.8.2 Operation

Higher layers of serial bus management configure a Multiaddress Talker as follows:

- 1) They set the Code Ports of the HEADER RECEIVERS to different values in the range 1-1022.
- 2) They make the ENABLE R input of each HEADER RECEIVER False.
- 3) For a Multiaddress On-Demand Talker, they load data into the DATA SENDER's Data Port and then make ENABLE S True. For a Multiaddress Transaction Talker, they reset the latch controlling ENABLE S, so that ENABLE S is False.

Operation of a Multiaddress Talker is similar to that described in Paragraph 6.1.4 for the corresponding type of (single address) Talker, except that the DATA SENDER is selected when any of the HEADER RECEIVERS matches its address in the S field of a Header. In Figure 6-8 above, the S SELECT input of the DATA SENDER is True whenever any one of the S SELECT outputs of the HEADER RECEIVERS is True. For a Multiaddress Transaction Talker, there is a single latch reflecting whether data has been provided by higher layers of serial bus management - its output is connected to the ENABLE S inputs of all the HEADER RECEIVERS.

6.1.9 Ecouteur multiadresse

Un écouteur multiadresse est représenté par la figure 6-9 ci-dessous. Il diffère des types d'écouteurs décrits précédemment par le fait que les sorties R SELECT de plusieurs RECEPTEURS D'EN-TETE sont câblées en OU logique pour fournir l'entrée R SELECT du RECEPTEUR DE DONNEE, et que les sorties R STROBE des RECEPTEURS D'EN-TETE sont câblées en OU logique pour fournir un signal qui annonce au RECEPTEUR DE DONNEE l'arrivée d'une nouvelle donnée.

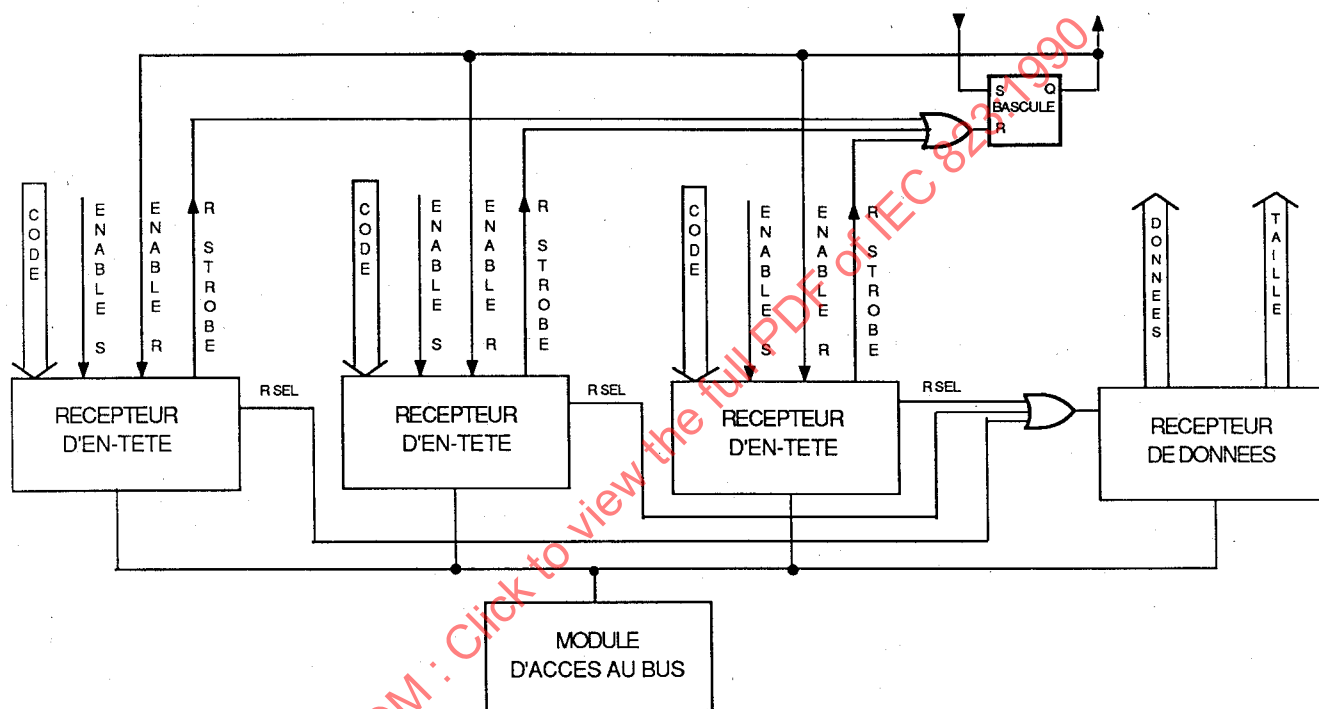


Figure 6-9 - Ecouteur multiadresse transactionnel

6.1.9.1 But

Un écouteur multiadresse est utile lorsqu'une destination particulière pour les trames de transfert de données peut être adressée de plusieurs manières. Par exemple, dans un système multiprocesseur, il peut être souhaitable d'envoyer un message ou une notification d'événement à "tous les processeurs", "tous les processeurs XX" ou un processeur particulier.

6.1.9 Multiaddress Listener

A Multiaddress Listener is shown in Figure 6-9. It differs from the types of Listeners described previously in that the R SELECT outputs of several HEADER RECEIVERS are logically OR'ed to produce the R SELECT input of the DATA RECEIVER, and that the R STROBE outputs of the HEADER RECEIVERS are logically OR'ed to produce a signal that announces the arrival of new data at the DATA RECEIVER.

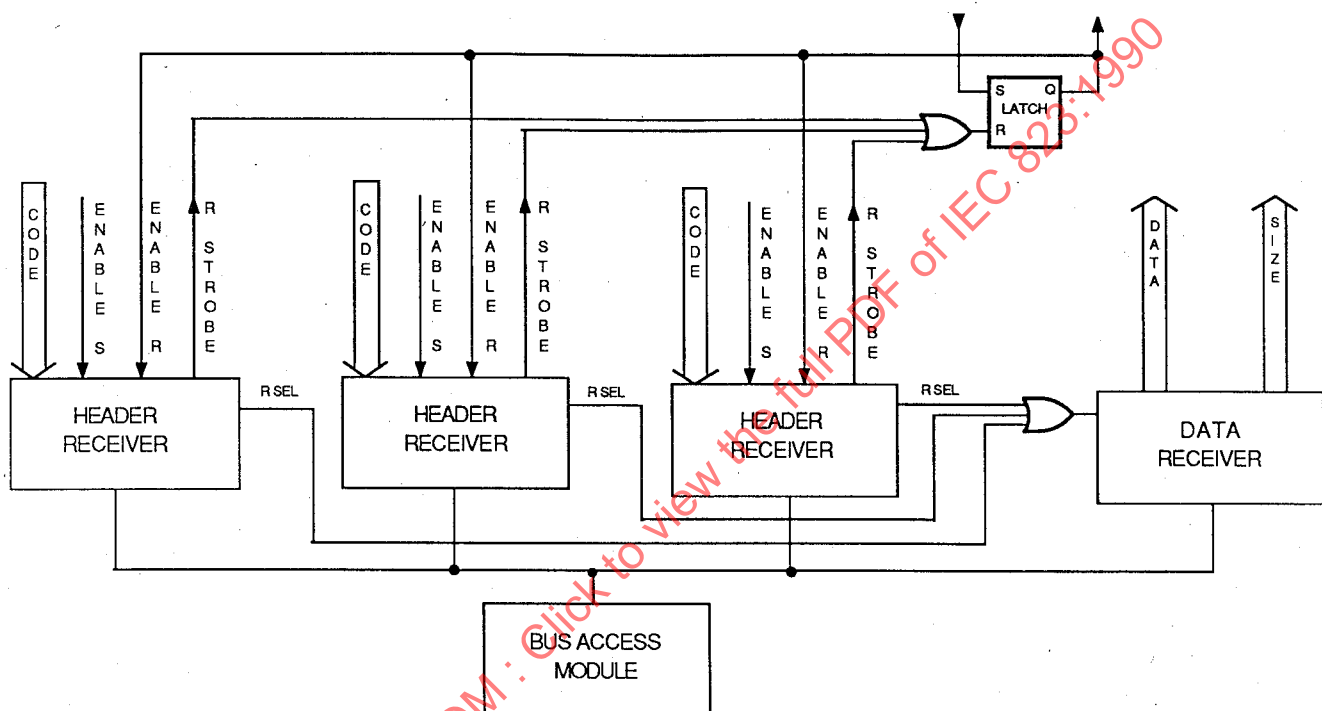


Figure 6-9 - Multiaddress Transaction Listener

6.1.9.1 Purpose

A Multiaddress Listener is useful whenever a particular destination for data transfer frames can be specified in any of several ways. For example, in a multiprocessor system it might be desirable to send a message or event notification to "all processors", "all XX processors", or a particular processor.

6.1.9.2 *Fonctionnement*

Les couches supérieures du gestionnaire du bus série configurent un écouteur multiadresse comme suit:

- 1) Elles positionnent les ports de code des RECEPTEURS D'EN-TETE à différentes valeurs comprises entre 1 et 1022.
- 2) Elles mettent l'entrée ENABLE S de chaque RECEPTEUR D'EN-TETE à l'état faux.
- 3) Elles mettent l'entrée ENABLE R de chaque RECEPTEUR D'EN-TETE à l'état vrai.

Le fonctionnement d'un écouteur multiadresse est semblable à celui du type correspondant d'écouteur (adresse simple) décrit aux paragraphes 6.1.5 ou 6.1.6, excepté que le RECEPTEUR DE DONNEE est sélectionné quand un quelconque des RECEPTEURS D'EN-TETE reconnaît son adresse dans le champ R d'un en-tête. Dans la figure 6-9, page 206, l'entrée R SELECT du RECEPTEUR DE DONNEE est à l'état vrai quand une quelconque des sorties R SELECT du RECEPTEUR D'EN-TETE est à l'état vrai. Pour un écouteur multiadresse transactionnel, il y a une simple bascule indiquant si la donnée reçue par le RECEPTEUR DE DONNEE a été traitée par les couches supérieures du gestionnaire du bus série - sa sortie est connectée aux entrées ENABLE R de tous les RECEPTEURS D'EN-TETE.

6.1.9.2 Operation

Higher layers of serial bus management configure a Multiaddress Listener as follows:

- 1) They set the Code Ports of the HEADER RECEIVERS to different values in the range 1-1022.
- 2) They make the ENABLE S input of each HEADER RECEIVER False.
- 3) They make the ENABLE R input of each HEADER RECEIVER True.

The operation of a Multiaddress Listener is similar to that of the corresponding type of (single-address) Listener as described in Paragraphs 6.1.5 or 6.1.6, except that the DATA RECEIVER is selected whenever any of the HEADER RECEIVERS matches its address in the R field of a Header. In Figure 6-9, page 207, the R SELECT input of the DATA RECEIVER is True when any one of the R SELECT outputs of the HEADER RECEIVERS is True. For a Multiaddress Transaction Listener, there is a single latch reflecting whether data received by the DATA RECEIVER has been processed by higher layers of serial bus management -- the latch's output is connected to the ENABLE R inputs of all the HEADER RECEIVERS.

IECNORM.COM :: Click to view the full PDF of IEC 823-1-990

6.1.10 Gestionnaire de priorité variable

Un gestionnaire de priorité variable est représenté par la figure 6-10 ci-dessous. Il diffère des autres gestionnaires par le fait que son port de priorité est connecté à un compteur 3 bits, lequel est ajusté selon les résultats de la transmission des trames pour maximiser l'égalité d'utilisation et la capacité de sortie du bus série.

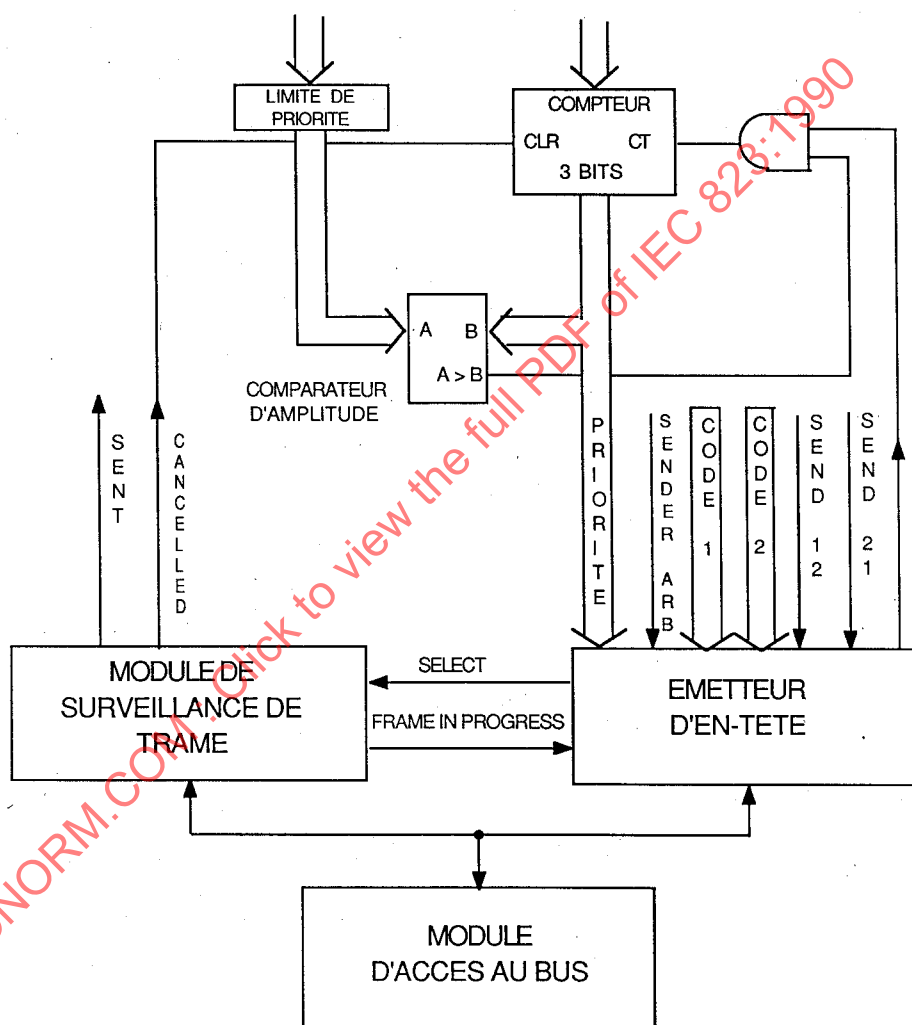


Figure 6-10 - Gestionnaire de priorité variable

6.1.10.1 But

Comme d'autre gestionnaires, ce groupe est utilisé pour déclencher une trame sur le bus série. Il diffère par le fait qu'il contient du matériel supplémentaire pour assurer que chaque trame du bus série obtienne un accès au bus série équitable et efficace.

6.1.10 Variable Priority Controller

A Variable Priority Controller is shown in Figure 6-10. It differs from other Controllers in that its Priority Port is connected to a 3-bit counter, which is adjusted according to frame transmission results to maximize serial bus "fairness" and throughput.

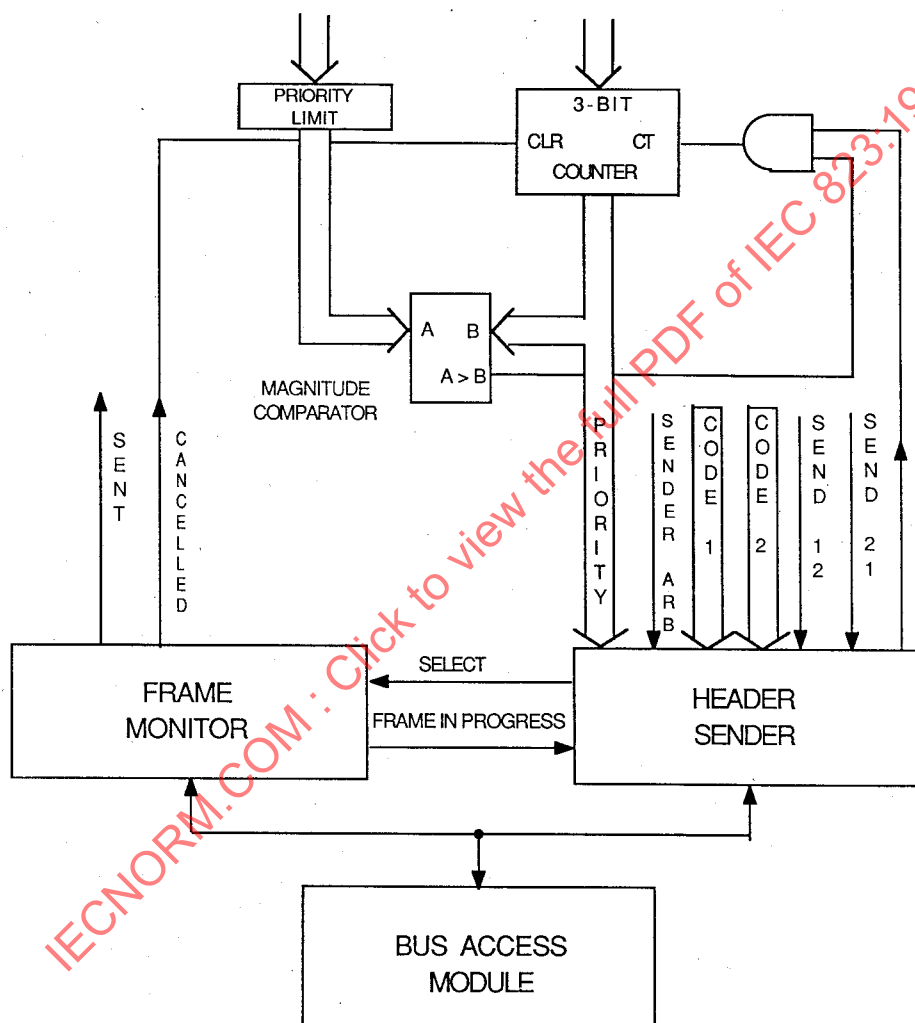


Figure 6-10 - Variable Priority Controller

6.1.10.1 Purpose

Like other Controllers, this group is used to initiate a frame on the serial bus. It differs in that it contains additional hardware to help ensure that each serial bus frame gets fair and efficient access to the serial bus.

6.1.10.2 Fonctionnement

Lorsqu'un en-tête doit être envoyé, les couches supérieures du gestionnaire du bus série configurent un gestionnaire de priorité variable comme suit:

- 1) Elles chargent une valeur initiale (reçue) du champ de priorité de la trame dans le compteur 3 bits connecté au port de priorité de l'EMETTEUR D'EN-TETE.
- 2) Elles chargent une valeur limite maximale pour la priorité de la trame dans les mémoires de limite de priorité.
- 3) Elles positionnent le port de code 1, le port de code 2 et le port DSAE de l'EMETTEUR D'EN-TETE avec les valeurs correspondantes.
- 4) Elles mettent à l'état vrai une des entrées SEND 12 et SEND 21 de l'EMETTEUR D'EN-TETE.

L'EMETTEUR D'EN-TETE commence alors à essayer d'envoyer l'en-tête. Il attend d'abord, si nécessaire, que le signal FRAME IN PROGRESS de son GESTIONNAIRE DE TRAME passe à l'état faux. Il envoie alors un bit_départ et commence à émettre l'en-tête.

S'il perd l'arbitrage du bus série, l'EMETTEUR D'EN-TETE envoie une impulsion sur le signal de sortie LOST ARB. Si la valeur dans le champ de priorité de 3 bits est déjà égale au niveau de priorité limite mémorisé, cette impulsion sur LOST ARB reste sans effet. Autrement, l'impulsion sur LOST ARB incrémente le compteur afin que, lorsque l'EMETTEUR D'EN-TETE essaiera d'envoyer à nouveau l'en-tête après la trame en cours, il le fasse avec une valeur du champ de priorité augmentée d'une unité par rapport à la valeur précédemment utilisée. Autrement dit, si le groupe perd l'arbitrage du bus série, sa chance de l'obtenir la prochaine fois est augmentée.

Si l'EMETTEUR D'EN-TETE obtient l'arbitrage du bus série, il envoie une impulsion sur son signal de sortie SELECT vers le GESTIONNAIRE DE TRAME, qui valide les sorties SENT et CANCELLED du GESTIONNAIRE DE TRAME. Si le GESTIONNAIRE DE TRAME détecte alors une valeur du type de trame de 111, il envoie une impulsion sur sa sortie CANCELLED. Cela remet à zéro le compteur 3 bits de priorité afin que, lorsque l'EMETTEUR D'EN-TETE essaiera d'envoyer à nouveau l'en-tête, après la trame en cours, il le fasse avec une valeur zéro du champ de priorité. Cela aide à éviter à la transmission (typiquement répétée) des trames annulées d'interférer avec le trafic utile du bus série.

(Lorsque l'EMETTEUR D'EN-TETE obtient l'arbitrage et la trame n'est pas annulée, l'opération est terminée et les trames suivantes sont transmises avec une nouvelle priorité initiale [reçue] comme décrit ci-dessus.)

OBSERVATION 6.2:

La caractéristique de priorité variable est indépendante des autres attributs des gestionnaires. Il est ainsi possible de réaliser des groupes plus complexes tels que les gestionnaires d'écriture à priorité variable.

6.1.10.2 Operation

When a Header is to be sent, higher layers of serial bus management configure a Variable Priority Controller as follows:

- 1) They load an initial (incoming) value for the Frame Priority field into the 3-bit counter that is connected to the Priority Port of the HEADER SENDER.
- 2) They load an upper-limit value for the Frame Priority into the Priority Limit latches.
- 3) They set the Code 1 Port, Code 2 Port and DSAE Port of the HEADER SENDER to appropriate values.
- 4) They make one of the SEND 12 and SEND 21 inputs of the HEADER SENDER True.

The HEADER SENDER then begins to try to send the Header. First, it waits, if necessary, for the FRAME IN PROGRESS signal from its FRAME MONITOR to go False. Then it sends a Start_bit and begins to send the Header.

If it loses the serial bus arbitration, the HEADER SENDER pulses the LOST ARB output. If the value in the 3-bit Priority value is already equal to the value in the Priority Limit latches, this pulse on LOST ARB has no effect. Otherwise the pulse on LOST ARB increments the counter so that, when the HEADER SENDER tries to send the Header again after the current frame, it will do so with a Priority field value one higher than the value it just used. In other words, if the group loses the serial bus arbitration, its chances to win it next time are increased.

If the HEADER SENDER wins the serial bus arbitration, it pulses its SELECT output to the FRAME MONITOR, which enables the FRAME MONITOR's SENT and CANCELLED outputs. If the FRAME MONITOR then monitors a Frame Type value of 111, it pulses its CANCELLED output. This clears the 3-bit Priority Counter, so that when the HEADER SENDER tries to send the Header again after the current frame, it will do so with a Priority field value of zero. This helps keep the (typically repeated) transmission of cancelled frames from interfering with more productive traffic on the serial bus.

(When the HEADER SENDER wins the arbitration, and the frame *is* not cancelled, the operation is over and future frames are transmitted with a new initial [incoming] priority as described above.)

OBSERVATION 6.2:

The Variable Priority characteristic is independent of other attributes of Controllers. Thus it is possible to implement more complex groups such as Variable Priority Writing Controllers.

6.2 Groupes composés

Les combinaisons suivantes de modules incluent les fonctions de plusieurs groupes simples décrits précédemment, mais interconnectés et utilisés comme une unité autonome par les couches supérieures du gestionnaire du bus série.

6.2.1 Gestionnaire d'écriture

Un gestionnaire d'écriture est représenté par la figure 6-11 ci-dessous. Il est composé simplement d'un gestionnaire et d'un parleur sur la même carte, avec l'adresse S envoyée par le gestionnaire servant à sélectionner le parleur.

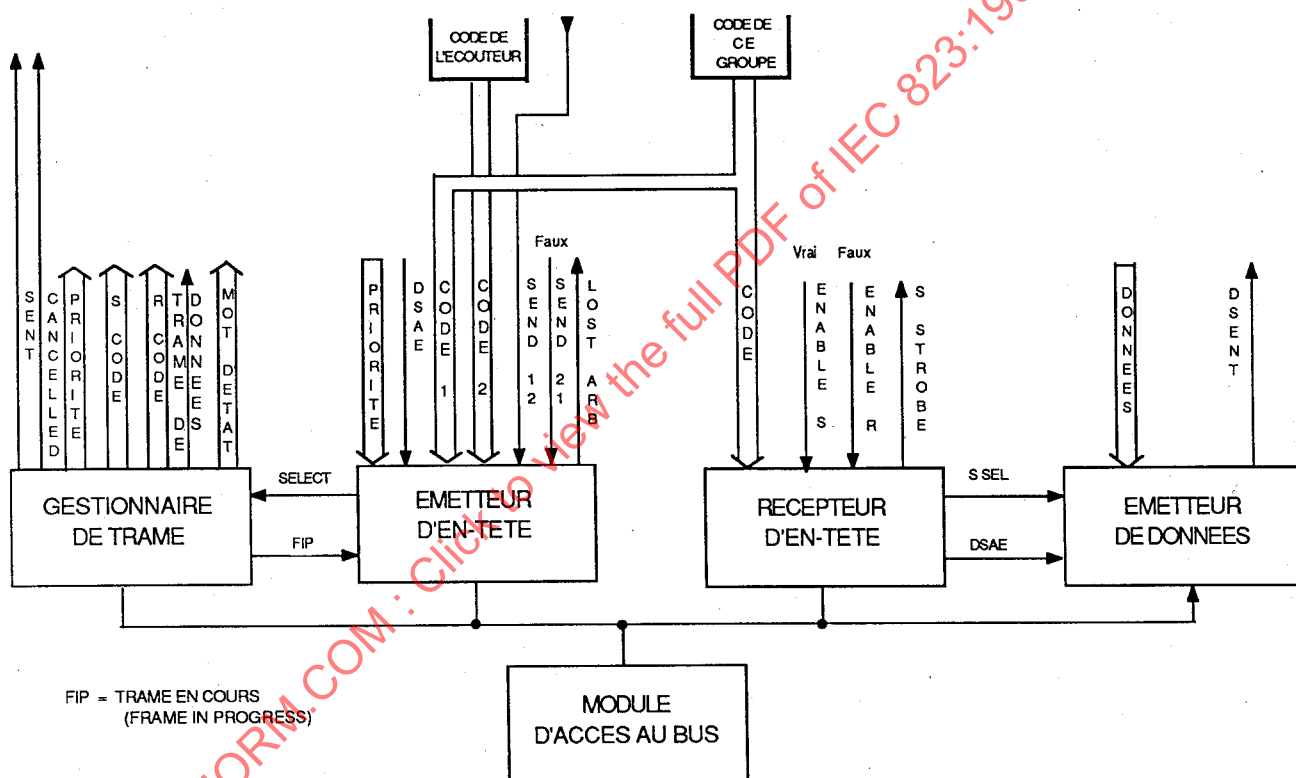


Figure 6-11 - Gestionnaire d'écriture

6.2.1.1 But

Les couches supérieures du gestionnaire du bus série peuvent utiliser un gestionnaire d'écriture pour envoyer des données à un écouteur sur le bus série, dans des applications où les données sont expédiées dès qu'elles deviennent disponibles.

6.2 Compound groups

The following combinations of modules include the functions of more than one of the Simple Groups described previously, but are interconnected and used as a single unit by higher layers of serial bus management.

6.2.1 Writing Controller

A Writing Controller is shown in Figure 6-11. It is simply composed of a Controller and a Talker on the same board, with the S address sent by the Controller serving to select the Talker.

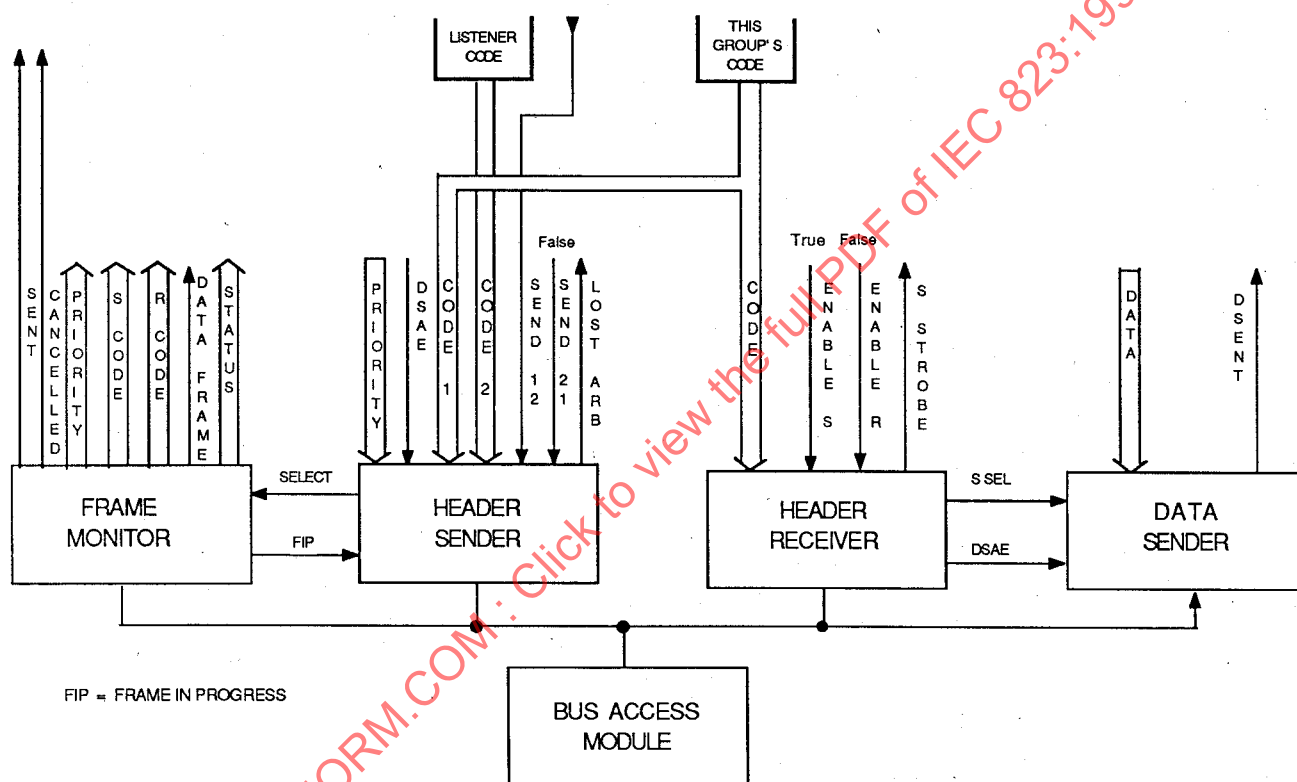


Figure 6-11 - Writing Controller

6.2.1.1 Purpose

Higher layers of serial bus management can use a Writing Controller to send data to a Listener on the serial bus, in applications in which data is sent whenever it becomes available.

6.2.1.2 Fonctionnement

Lorsque les couches supérieures doivent envoyer des données à un écouteur, elles configurent le gestionnaire d'écriture comme suit:

- 1) Elles positionnent les ports de priorité et d'arbitrage de l'émetteur de données de l'EMETTEUR D'EN-TETE correspondant à la trame à envoyer.
- 2) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE avec l'adresse affectée à ce parleur.
- 3) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE avec l'adresse affectée à l'écouteur auquel la donnée doit être envoyée.
- 4) Elles présentent les données à envoyer au port de données de l'EMETTEUR DE DONNEES.
- 5) Elles mettent à l'état vrai l'entrée ENABLE S du RECEPTEUR D'EN-TETE et à l'état faux son entrée ENABLE R.
- 6) Elles mettent à l'état vrai l'entrée SEND 12 de l'EMETTEUR D'EN-TETE, ce qui lui commande de déclencher la trame en envoyant son en-tête.

L'EMETTEUR D'EN-TETE commence alors à essayer d'envoyer un en-tête avec l'adresse du parleur dans le champ S et l'adresse de l'écouteur de destination dans le champ R. Il attend d'abord (si nécessaire) que son GESTIONNAIRE DE TRAME signale qu'aucune trame n'est en cours sur le bus série. Il envoie alors un bit_départ et commence son en-tête. Le gestionnaire peut obtenir les résultats suivants dans la tentative de transmission:

- 1) Il peut perdre l'arbitrage du bus série. Dans ce cas, il essaie simplement d'envoyer à nouveau l'en-tête après la fin de l'autre trame.
- 2) Il peut obtenir l'arbitrage mais avoir la trame annulée parce que l'écouteur n'est pas prêt pour la trame, c'est-à-dire parce que les couches supérieures du gestionnaire du bus série n'ont pas encore traité les données reçues dans une trame précédente. Dans ce cas, le gestionnaire d'écriture rapporte la situation aux couches supérieures du gestionnaire du bus série.
- 3) Il peut obtenir l'arbitrage et sélectionner son parleur, qui commence à envoyer les données. Cela est le cas normal.

Dans le cas normal, le gestionnaire d'écriture du RECEPTEUR D'EN-TETE détecte son adresse dans le champ S de l'en-tête et positionne à l'état vrai sa sortie S SELECT, qui est l'entrée S SELECT de l'EMETTEUR DE DONNEES. Entre-temps, un ou plusieurs RECEPTEURS D'EN-TETE dans les groupes écouteurs, ailleurs sur le bus série, sont sélectionnés par l'adresse R et font passer à l'état vrai leurs sorties R SELECT qui sont les entrées R SELECT de leurs RECEPTEURS DE DONNEES.

6.2.1.2 Operation

When higher layers need to send data to a Listener, they configure a Writing Controller as follows:

- 1) They set the HEADER SENDER's Priority and Data Sender Arbitration Ports as appropriate for the frame to be sent.
- 2) They set the HEADER SENDER's Code 1 Port and the HEADER RECEIVER's Code Port to the address assigned to this Talker.
- 3) They set the HEADER SENDER's Code 2 Port to the address assigned to the Listener to which the data is to be sent.
- 4) They present the data to be sent at the Data Port of the DATA SENDER.
- 5) They make the HEADER RECEIVER's ENABLE S input True and its ENABLE R input False.
- 6) They make the HEADER SENDER's SEND 12 input True, which commands it to initiate the frame by sending its Header.

The HEADER SENDER then begins to try to send a Header with this Talker's address in the S field and the address of the destination Listener in the R field. First, it waits (if necessary) for its FRAME MONITOR to signal that no serial bus frame is in progress. Then it sends a Start_bit and begins its Header. The Controller may encounter the following results in the transmission attempt:

- 1) It may lose the serial bus arbitration. In this case, it merely retries the attempt to send the Header after the other frame is over.
- 2) It may win the arbitration but have the frame cancelled because the Listener is not ready for the frame, i.e. because its higher layers of serial bus management have not yet processed data received in a previous frame. In this case, the Writing Controller reports the situation to higher layers of serial bus management.
- 3) It may win the arbitration and select its Talker, which proceeds to send the data. This is the normal case.

In the normal case, the Writing Controller's HEADER RECEIVER detects its address in the S field of the Header, and makes its S SELECT output True, which is the S SELECT input of the DATA SENDER. Meanwhile, one or more HEADER RECEIVERS in Listener groups, elsewhere on the serial bus, are selected by the R address and make their R SELECT outputs True, which are the R SELECT inputs of their DATA RECEIVERS.

L'EMETTEUR DE DONNEES positionne une des valeurs 001-110 dans le champ de type de trame qui suit l'en-tête pour indiquer la quantité de données qu'il doit envoyer. Il envoie ensuite cette quantité de données. L'EMETTEUR DE DONNEES envoie alors 010 dans le champ suivant de mot d'état de la trame tandis que le RECEPTEUR DE DONNEES envoie 001. Ces valeurs sont reliées en OU logique sur le bus série pour donner le mot d'état de la trame résultant de 011. La validation simultanée de SENT par le GESTIONNAIRE DE TRAME et de S STROBE par le RECEPTEUR D'EN-TETE informe les couches supérieures du gestionnaire du bus série que le mot d'état de la trame est correct et que les données ont été envoyées. Entre-temps, la sortie R STROBE du RECEPTEUR D'EN-TETE de l'écouteur informe ses couches supérieures que les données ont été reçues et il convient qu'elles soient lues dans le RECEPTEUR DE DONNEES.

6.2.2 Gestionnaire de lecture

Un gestionnaire de lecture est représenté par la figure 6-12 ci-dessous. Il est composé simplement d'un gestionnaire et d'un écouteur sur la même carte avec l'adresse R envoyée par le gestionnaire servant à sélectionner l'écouteur.

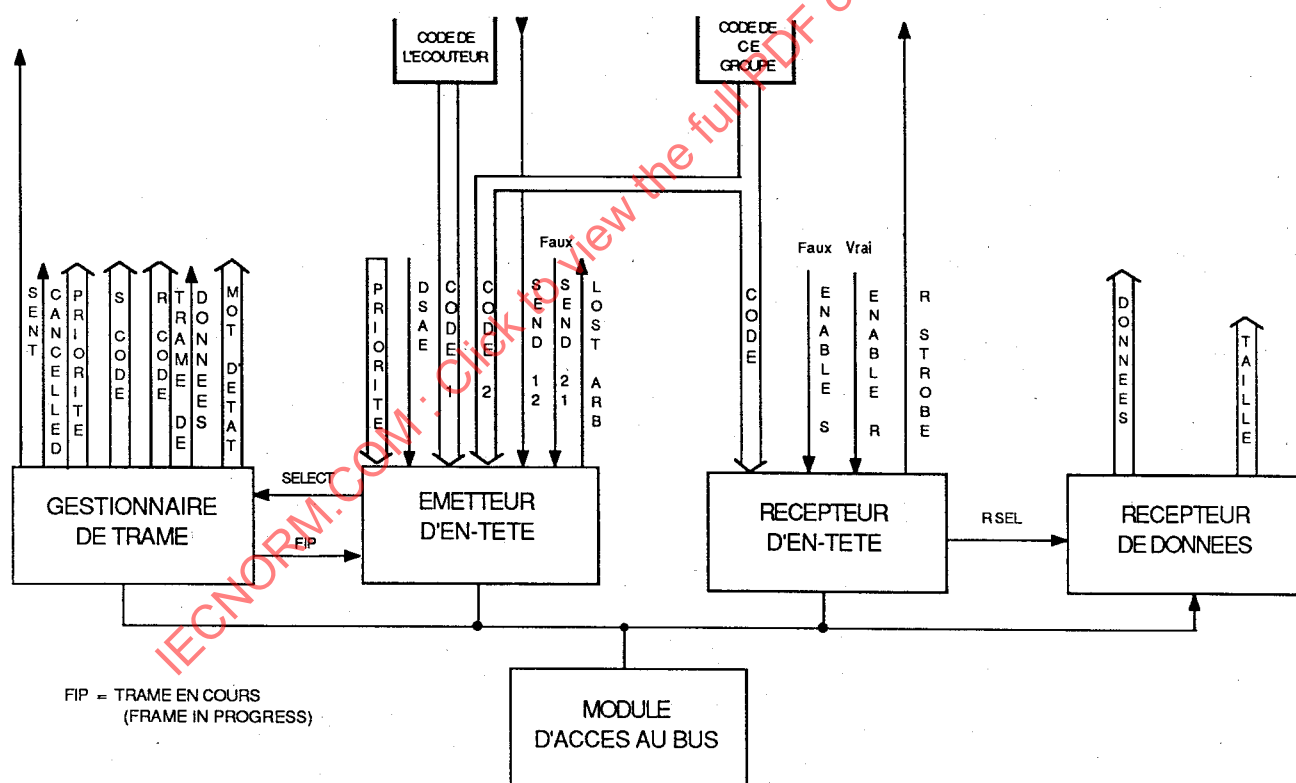


Figure 6-12 - Gestionnaire de lecture

6.2.2.1 But

Les couches supérieures du gestionnaire du bus série peuvent utiliser un gestionnaire de lecture pour lire les données à partir d'un parleur sur le bus série, dans les applications où les données sont lues chaque fois qu'on en a besoin.

The DATA SENDER drives one of the values 001-110 in the Frame Type field that follows the Header to indicate how much data it has to send. It then goes on to send that amount of data. The DATA SENDER then sends 010 in the following Frame Status field while the DATA RECEIVER sends 001. These values are OR'ed by the serial bus to give the result Frame Status of 011. Simultaneous assertion of SENT by the FRAME MONITOR and S STROBE by the HEADER RECEIVER informs higher layers of serial bus management that the Frame Status is correct and that the data has been sent. Meanwhile, the R STROBE output of the Listener's HEADER RECEIVER informs its higher layers that data has been received and should be read from its DATA RECEIVER.

6.2.2 Reading Controller

A Reading Controller is shown in Figure 6-12. It is simply composed of a Controller and a Listener on the same board, with the R address sent by the Controller serving to select the Listener.

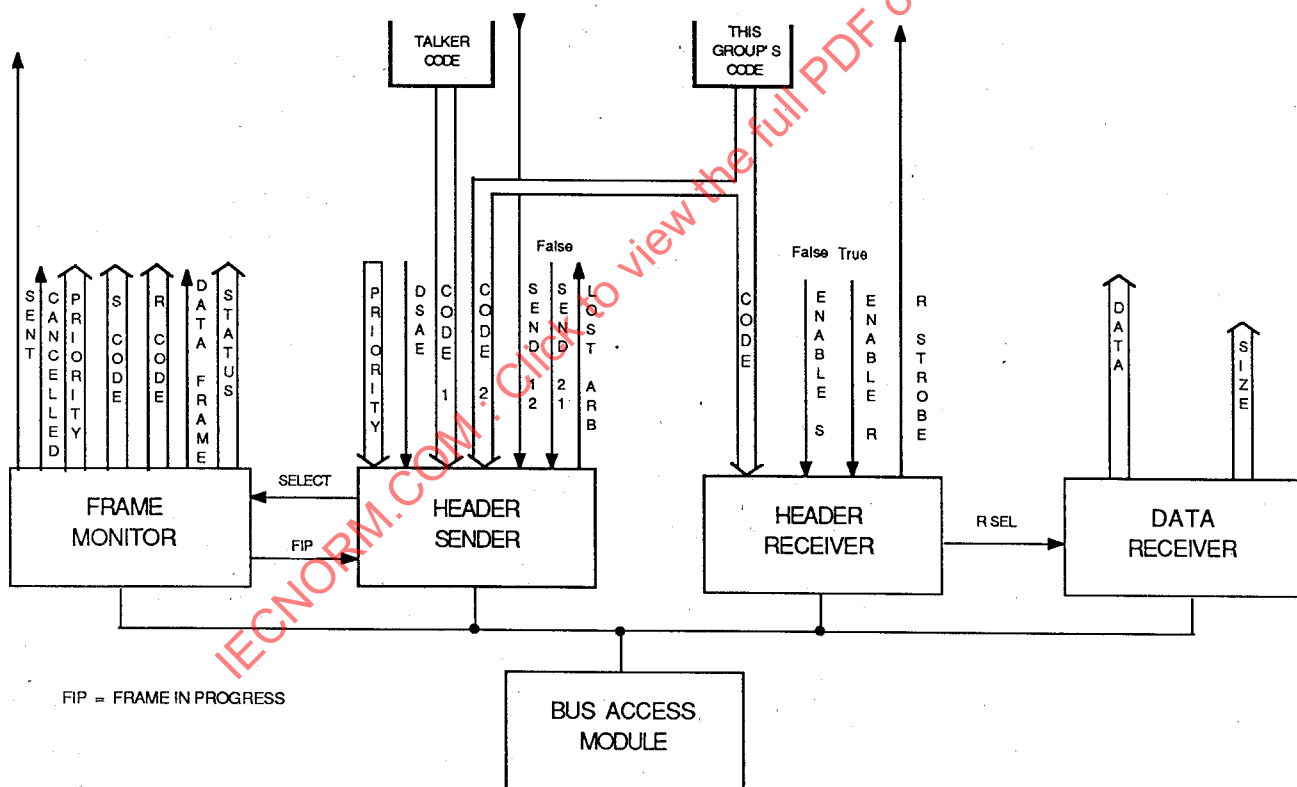


Figure 6-12 - Reading Controller

6.2.2.1 Purpose

Higher layers of serial bus management can use a Reading Controller to read out data from a Talker on the serial bus, in applications in which data is read out whenever it is needed.

6.2.2.2 Fonctionnement

Lorsque les couches supérieures doivent lire les données à partir d'un parleur, elles configurent le gestionnaire de lecture comme suit:

- 1) Elles positionnent les ports de priorité et les ports d'arbitrage de l'émetteur de données de l'EMETTEUR D'EN-TETE selon l'opération à effectuer.
- 2) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE avec l'adresse affectée à cet écouteur.
- 3) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE avec l'adresse affectée au parleur à partir duquel les données doivent être lues.
- 4) Elles mettent à l'état vrai l'entrée ENABLE R du RECEPTEUR D'EN-TETE et à l'état faux son entrée ENABLE S.
- 5) Elles mettent à l'état vrai l'entrée SEND 12 de l'EMETTEUR D'EN-TETE, ce qui lui commande de déclencher la trame en envoyant son en-tête.

L'EMETTEUR D'EN-TETE commence alors à essayer d'envoyer un en-tête avec l'adresse de cet écouteur dans le champ R et l'adresse du parleur considéré dans le champ S. Il attend d'abord (si nécessaire) que son GESTIONNAIRE DE TRAME signale qu'aucune trame n'est en cours d'envoi sur le bus série. Il envoie ensuite un bit_départ et commence son en-tête. Le gestionnaire peut rencontrer les résultats suivants dans la tentative de transmission:

- 1) Il peut perdre l'arbitrage du bus série. Dans ce cas, il essaie simplement d'envoyer à nouveau l'en-tête lorsque l'autre trame est terminée.
- 2) Il peut obtenir l'arbitrage mais peut avoir la trame annulée parce que le parleur n'est pas prêt pour cette trame, c'est-à-dire parce que ses couches supérieures du gestionnaire du bus série n'ont pas encore fourni les données à envoyer dans cette trame. Dans ce cas, le gestionnaire d'écriture rapporte la situation aux couches supérieures du gestionnaire du bus série.
- 3) Il peut obtenir l'arbitrage et sélectionner le parleur choisi, qui commence à envoyer les données. C'est le cas normal.

Dans le cas normal, le RECEPTEUR D'EN-TETE du gestionnaire de lecture est sélectionné par l'adresse R de l'en-tête et met à l'état vrai sa sortie R SELECT qui correspond à l'entrée R SELECT du RECEPTEUR DE DONNEES. Entre-temps, un ou plusieurs RECEPTEURS D'EN-TETE des groupes parleurs, ailleurs sur le bus série, sont sélectionnés par l'adresse S et mettent au niveau vrai leurs sorties S SELECT qui correspondent aux entrées S SELECT de leurs EMETTEURS DE DONNEES.

6.2.2.2 Operation

When higher layers need to read out data from a Talker, they configure a Reading Controller as follows:

- 1) They set the HEADER SENDER's Priority and Data Sender Arbitration Ports as appropriate for the operation to be performed.
- 2) They set the HEADER SENDER's Code 2 Port and the HEADER RECEIVER's Code Port to the address assigned to this Listener.
- 3) They set the HEADER SENDER's Code 1 Port to the address assigned to the Talker from which the data is to be read out.
- 4) They make the HEADER RECEIVER's ENABLE R input True, and its ENABLE S input False.
- 5) They make the HEADER SENDER's SEND 12 input True, which commands it to initiate the frame by sending its Header.

The HEADER SENDER then begins to try to send a Header with this Listener's address in the R field and the address of the intended Talker in the S field. First, it waits (if necessary) for its FRAME MONITOR to signal that no serial bus frame is in progress. Then it sends a Start_bit and begins its Header. The Controller may encounter the following results in the transmission attempt:

- 1) It may lose the serial bus arbitration. In this case, it merely retries the attempt to send the Header after the other frame is over.
- 2) It may win the arbitration but have the frame cancelled because the Talker is not ready for the frame, i.e. because its higher layers of serial management have not yet provided data to send in this frame. In this case, the Writing Controller reports the situation to higher layers of serial bus management.
- 3) It may win the arbitration and select the intended Talker, which proceeds to send data. This is the normal case.

In the normal case, the Reading Controller's HEADER RECEIVER is selected by the R address in the Header and makes its R SELECT output True, which is the R SELECT input of the DATA RECEIVER. Meanwhile, one or more HEADER RECEIVERS in Talker groups, elsewhere on the serial bus is/are selected by the S address, and make their S SELECT outputs True, which are the S SELECT inputs of their DATA SENDERS.

L'EMETTEUR ou les EMETTEURS DE DONNEES positionnent une des valeurs 001-110 dans le champ de type de trame qui suit l'en-tête pour indiquer la quantité des données à envoyer, suivi par cette quantité de données. Les données sont captées par le RECEPTEUR DE DONNEES du gestionnaire d'écriture. L'EMETTEUR DE DONNEES envoie ensuite 010 dans le champ de mot d'état de la trame tandis que le RECEPTEUR DE DONNEES envoie 001. Une opération de OU logique sur ces valeurs par le bus série donne un mot d'état de la trame résultant de 011. La validation simultanée de SENT par le GESTIONNAIRE DE TRAME et de R STROBE par le RECEPTEUR D'EN-TETE informe les couches supérieures du gestionnaire du bus série que le mot d'état de la trame est correct et qu'ils peuvent lire les données à partir des RECEPTEUR DE DONNEES.

6.2.3 Emetteur-récepteur de bus virtuel

L'émetteur-récepteur de bus virtuel (VBT) est représenté par la figure 6-13 ci-dessous. Il est composé d'un gestionnaire d'écriture et d'un écouteur transactionnel, mais l'EMETTEUR DE DONNEES et le RECEPTEUR DE DONNES sont reliés au même RECEPTEUR D'EN-TETE et répondent ainsi à la même adresse.

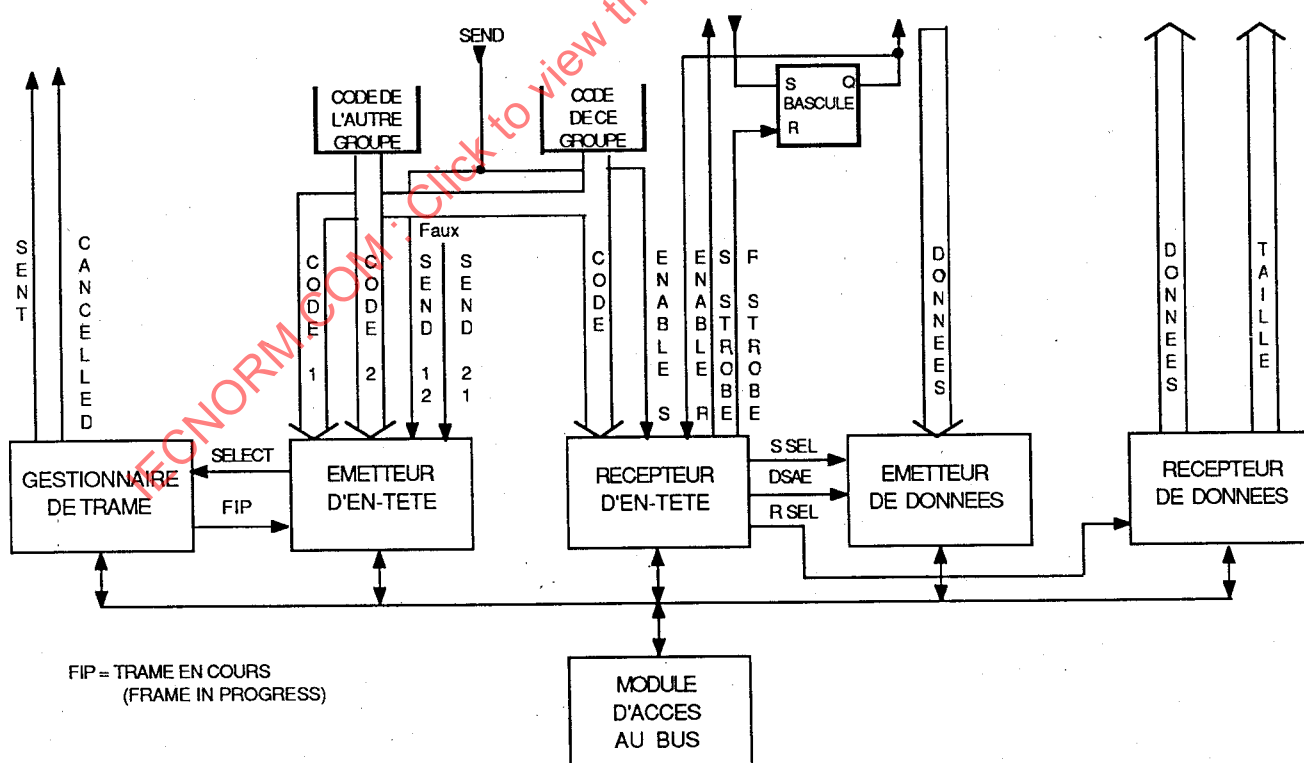


Figure 6-13 - Emetteur-récepteur de bus virtuel

The DATA SENDER(S) drive one of the values 001-110 in the Frame Type field that follows the Header, to indicate how much data will be sent, followed by that amount of data. The data is captured by the Writing Controller's DATA RECEIVER. The DATA SENDER then sends 010 in the Frame Status field while the DATA RECEIVER sends 001. These values are OR'ed by the serial bus to give the result Frame Status of 011. Simultaneous assertion of SENT by the FRAME MONITOR and R STROBE by the HEADER RECEIVER informs higher layers of serial bus management that the Frame Status is correct and that they can read the data from the DATA RECEIVER.

6.2.3 Virtual Bus Transceiver

A Virtual Bus Transceiver (VBT) is shown in Figure 6-13. It is composed of a Writing Controller and a Transaction Listener, except that the DATA SENDER and DATA RECEIVER are both connected to the same HEADER RECEIVER, so that they respond to the same address.

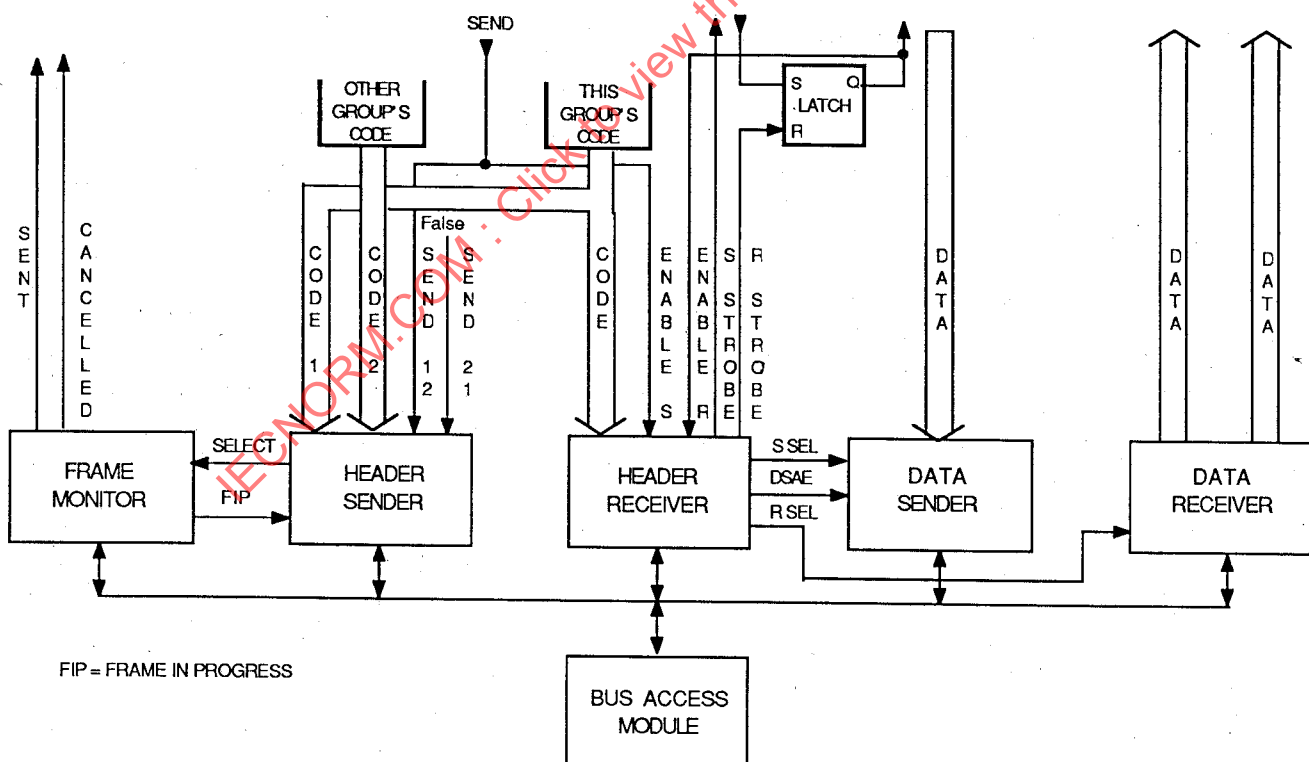


Figure 6-13 - Virtual Bus Transceiver

6.2.3.1 But

Un nombre d'émetteurs-récepteurs de bus virtuel peuvent être utilisés pour réaliser un "bus virtuel", à l'aide duquel des cartes peuvent échanger des données comme sur un bus parallèle de fond de panier. Par exemple, un VBT sur une carte processeur pourrait être utilisé pour envoyer une "commande" à un VBT sur une carte entrée/sortie. Lorsque l'opération commandée est achevée, le VBT de la carte entrée/sortie pourrait alors être utilisé pour envoyer un "résultat" en retour au VBT de la carte processeur.

6.2.3.2 Description

Lorsque le gestionnaire du VBT envoie un en-tête, il envoie l'adresse du VBT dans le champ S et l'EMETTEUR DE DONNEES envoie les données à un autre groupe. Lorsqu'un autre groupe envoie des données dans une trame qui possède cette adresse de groupe dans le champ R, le RECEPTEUR DE DONNEES saisit les données. Ainsi, ce groupe opère de façon semblable à un couplage d'un maître et d'un esclave d'un bus parallèle. Comme pour un bus de fond de panier avec plusieurs maîtres, le temps pour obtenir le contrôle d'un bus virtuel peut varier selon le trafic sur le bus et la priorité du "maître". Un bus virtuel a, comparativement à un bus parallèle, un "temps de propagation" plus long pour transférer réellement les données. Cependant, si le bus parallèle et le bus série sont tous les deux utilisés dans le même système, et que le bus parallèle est fortement chargé tandis que le bus série ne l'est pas, le transfert des données sur le bus série peut en réalité être plus rapide.

6.2.3.3 Fonctionnement

Quand un bus virtuel est initialisé, les couches supérieures du gestionnaire du bus série configurent un émetteur-récepteur de bus virtuel comme suit:

- 1) Elles positionnent à la fois le port du code 1 de l'EMETTEUR D'EN-TETE et le port du code du RECEPTEUR D'EN-TETE à l'adresse attribuée à ce groupe.
- 2) Elles positionnent le port d'arbitrage de l'émetteur de données de l'EMETTEUR D'EN-TETE à l'état vrai.
- 3) Supposant qu'il n'y a pas encore de données à envoyer, elles mettent le signal SEND à l'état faux. Cela met à la fois l'entrée SEND 12 de l'EMETTEUR D'EN-TETE et l'entrée ENABLE S du RECEPTEUR D'EN-TETE à l'état faux.
- 4) Elles positionnent la bascule de la figure 6-13, page 222. Cela met l'entrée ENABLE R du RECEPTEUR D'EN-TETE à l'état vrai.

6.2.3.1 Purpose

A number of Virtual Bus Transceivers can be used to implement a "virtual bus", by means of which boards can interchange data as on a parallel backplane bus. For example, a VBT on a processor board might be used to send a "command" to a VBT on an I/O board. When the commanded operation is complete, the VBT on the I/O board could then be used to send a "result" back to the processor's VBT.

6.2.3.2 Description

When the VBT's Controller sends a Header, it sends the VBT's address in the S field and the DATA SENDER sends data to another group. When another group sends data in a frame that has this group's address in the R field, the DATA RECEIVER captures the data. Thus, this group operates similarly to the combination of a parallel bus Master and Slave. Like a backplane bus with multiple Masters, the time to gain control of a virtual bus may vary based on bus traffic and the priority of the "Master". A virtual bus differs from a parallel bus in that it has a longer "propagation time" to actually transfer the data. However, if both a parallel bus and a serial bus are used in the same system, and the parallel bus is heavily loaded while the serial bus is not, transferring data on the serial bus may actually be faster.

6.2.3.3 Operation

When a virtual bus is set up, higher layers of serial bus management configure a Virtual Bus Transceiver as follows:

- 1) They set both the HEADER SENDER's Code 1 Port and the HEADER RECEIVER's Code Port to the address assigned to this group.
- 2) They set the HEADER SENDER's Data Sender Arbitration Port to True.
- 3) Assuming there is no data yet to be sent, they make the SEND signal False. This makes both the HEADER SENDER's SEND 12 input and the HEADER RECEIVER's ENABLE S input False.
- 4) They set the latch in Figure 6-13, page 223. This makes the HEADER RECEIVER's ENABLE R input True.

Quand une donnée doit être envoyée vers un autre groupe, les couches supérieures procèdent comme suit:

- 1) Elles positionnent le port de priorité de l'EMETTEUR D'EN-TETE d'une manière appropriée pour la trame à émettre.
- 2) Elles envoient le port de code 2 de l'EMETTEUR D'EN-TETE à l'adresse attribuée à l'émetteur-récepteur de bus virtuel (ou autre écouteur) auquel la donnée doit être envoyée.
- 3) Elles présentent les données à envoyer sur le port de données de l'EMETTEUR DE DONNEES.
- 4) Elles mettent le signal SEND à l'état vrai, ce qui a pour conséquence de mettre l'entrée ENABLE S du RECEPTEUR D'EN-TETE et l'entrée SEND 12 de l'EMETTEUR D'EN-TETE à l'état vrai. Cela commande à l'EMETTEUR D'EN-TETE de déclencher la trame en envoyant son en-tête.

Par la suite, les opérations se poursuivent comme décrit pour le gestionnaire d'écriture (voir paragraphe 6.2.1). En résumé, le gestionnaire essaie d'envoyer l'en-tête jusqu'à ce qu'il obtienne l'arbitrage bus série. Si la trame est annulée, il en rend compte aux couches supérieures. Autrement, l'EMETTEUR D'EN-TETE envoie les données et les résultats sont transmis aux couches supérieures.

Si un autre groupe émetteur-récepteur de bus virtuel (ou d'autre gestionnaire) envoie un en-tête avec son adresse de groupe dans le champ R, les opérations se poursuivent comme décrit pour un écouteur de transfert (voir paragraphe 6.1.6). En résumé, si les données de la trame précédente n'ont pas encore été traitées par les couches supérieures, le RECEPTEUR D'EN-TETE annule la trame. Autrement, le RECEPTEUR DE DONNEES saisit les données et le RECEPTEUR D'EN-TETE notifie aux couches supérieures que les données ont été reçues en remettant à zéro la bascule. Quand les couches supérieures ont eu les données, elles positionnent la bascule pour revalider plus tard la réception des données.

When data is to be sent to another group, higher layers proceed as follows:

- 1) They set the HEADER SENDER's Priority Port as appropriate for the frame to be sent.
- 2) They set the HEADER SENDER's Code 2 Port to the address assigned to the Virtual Bus Transceiver (or other Listener) to which the data is to be sent.
- 3) They present the data to be sent at the Data Port of the DATA SENDER.
- 4) They make the SEND signal True, which makes the HEADER RECEIVER's ENABLE S input and the HEADER SENDER's SEND 12 input True. This commands the HEADER SENDER to initiate the frame by sending its Header.

Thereafter, operation proceeds as described for a Writing Controller (see Paragraph 6.2.1). In summary, the Controller tries to send the Header until it wins the serial bus arbitration. If the frame is cancelled, it reports this to higher layers. Otherwise, the DATA SENDER sends the data and the results are reported to higher layers.

If another Virtual Bus Transceiver group (or other Controller) sends a Header with this group's address in the R field, operation proceeds as described for a Transaction Listener (see Paragraph 6.1.6). In summary, if data from a previous frame has not yet been processed by higher layers, the HEADER RECEIVER cancels the frame. Otherwise, the DATA RECEIVER captures the data, and the HEADER RECEIVER notifies higher layers that data has arrived, by clearing the latch. When higher layers have read out the data, they set the latch to re-enable further data reception.

6.2.4 Sémaphore

Un groupe sémaphore est représenté par la figure 6-14 ci-dessous. Il est composé d'un gestionnaire, d'un indicateur, et des circuits logiques disposés pour fournir un mécanisme d'exclusion mutuelle par simple bit.

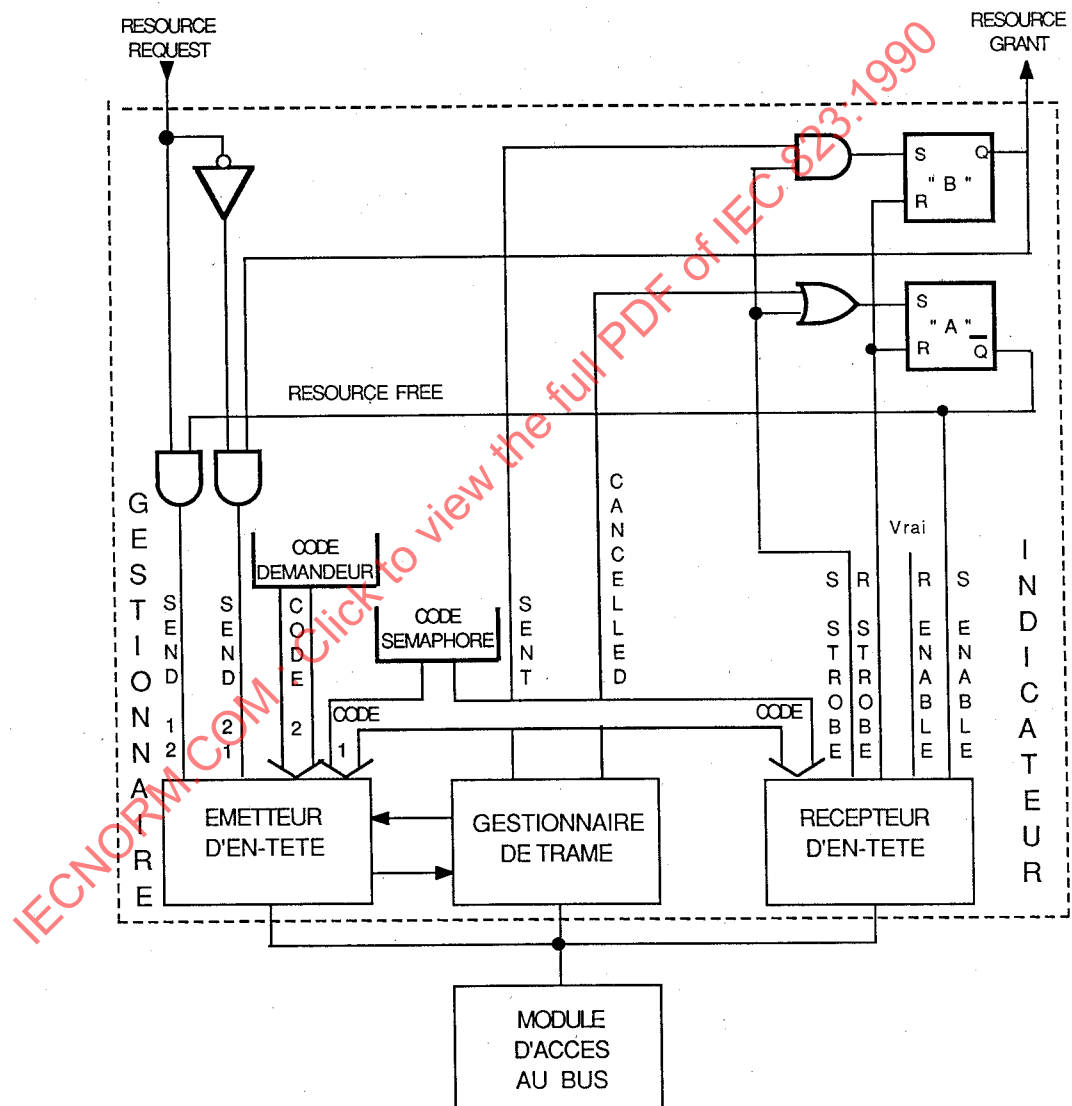


Figure 6-14 - Sémaphore

6.2.4 Semaphore

A Semaphore group is shown in Figure 6-14. It is composed of a Controller, a Flag and other logic arranged to implement a single-bit mutual exclusion mechanism.

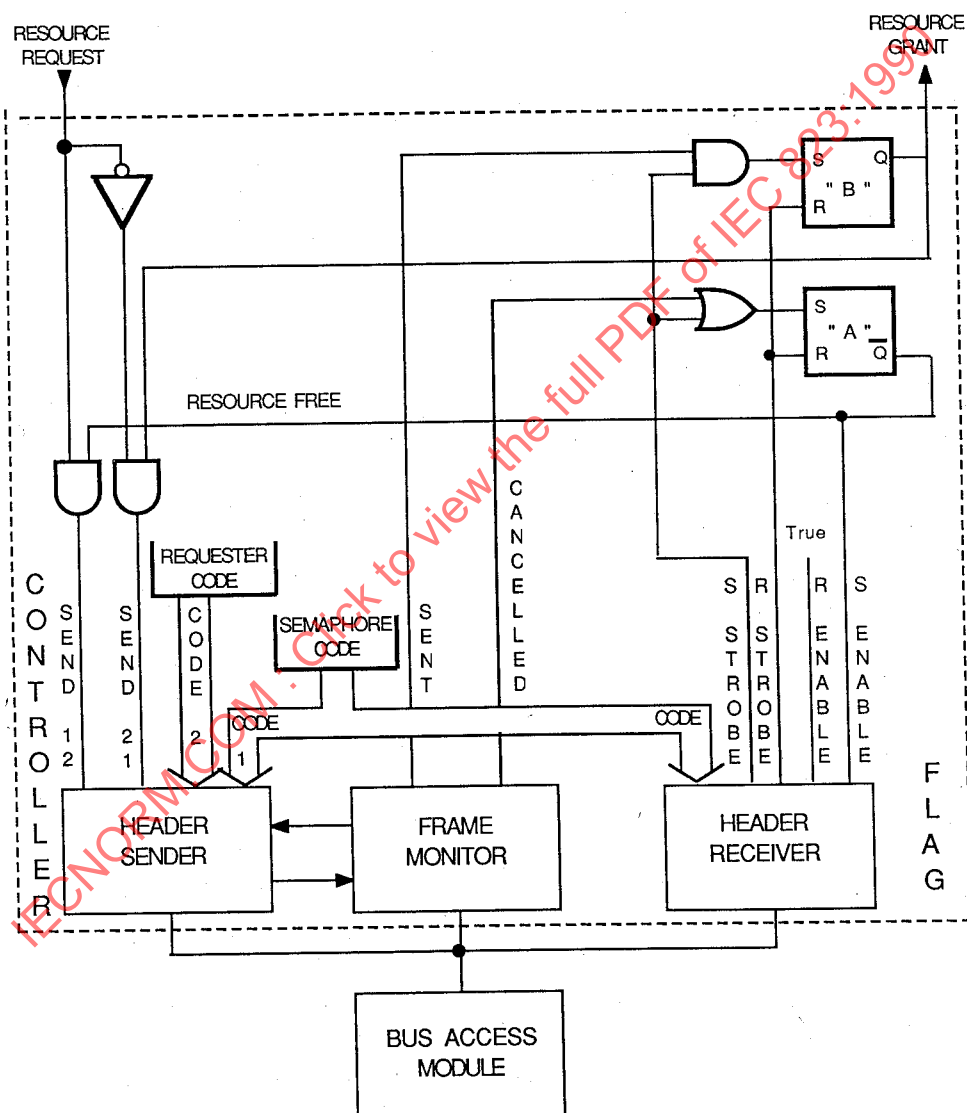


Figure 6-14 - Semaphore

6.2.4.1 But

Un groupe sémaphore est utilisé par les couches supérieures du gestionnaire du bus série pour obtenir l'usage exclusif de certaines ressources partageables du système et ensuite les libérer. C'est un groupe composé car le gestionnaire donne les moyens de mettre à un ou à zéro un bit qui contrôle l'accès à la ressource, tandis que son indicateur positionne le bit probablement "en parallèle avec" les indicateurs des autres groupes sémaphore.

L'état de l'indicateur représente l'état de la ressource partageable. Quand l'indicateur est positionné, la ressource est allouée/assignée au processeur ou à la tâche qui l'a positionné. Quand l'indicateur est positionné à zéro, les couches supérieures du gestionnaire du bus série peuvent utiliser le gestionnaire pour positionner l'indicateur et ainsi prendre le contrôle de la ressource.

6.2.4.2 Description

Dans la figure 6-14, page 228, la partie indicateur du groupe sémaphore comprend le RECEPTEUR D'EN-TETE et la bascule "A". La sortie "Q-inverse" de la bascule "A" est appelé le signal RESOURCE FREE.

Cet indicateur a plusieurs similitudes avec l'indicateur simple décrit au paragraphe 6.1.1:

- 1) Quand le RECEPTEUR D'EN-TETE est sélectionné par l'adresse S d'une trame réussie sur le bus série, il met sa sortie STROBE S à l'état vrai, ce qui positionne la bascule "A" et met RESOURCE FREE à l'état faux.
- 2) Quand le RECEPTEUR D'EN-TETE est sélectionné par l'adresse R d'une trame sur le bus série, il met sa sortie R STROBE à l'état vrai, ce qui remet à zéro la bascule "A" et met RESOURCE FREE à l'état vrai.
- 3) L'entrée ENABLE R du RECEPTEUR D'EN-TETE est vraie en permanence, si bien que le RECEPTEUR D'EN-TETE n'annule jamais une trame qui a son adresse dans le champ R.

Mais cet indicateur est différent d'un indicateur simple sur deux points:

- 1) La sortie RESOURCE FREE de l'indicateur est liée à l'entrée ENABLE S du RECEPTEUR D'EN-TETE du gestionnaire. En conséquence, si le RECEPTEUR D'EN-TETE est sélectionné par l'adresse S d'une trame sur le bus série, et que la bascule "A" est déjà positionnée, le RECEPTEUR D'EN-TETE annule la trame.
- 2) La sortie CANCELLED du GESTIONNAIRE DE TRAME est reliée en OU logique avec S STROBE pour positionner l'entrée de la bascule "A" à l'état vrai. Si un autre indicateur possède aussi le bit de sémaphore et est déjà positionné, il annulera la trame. Dans ce cas, cet indicateur est positionné pour être en accord avec l'autre indicateur. (Cette propriété suppose que si un module annule une trame avec une adresse de sémaphore dans le champ S, une trame sera envoyée avec l'adresse de sémaphore dans le champ R pour remettre à zéro l'indicateur et permettre de réémettre la trame.)

6.2.4.1 Purpose

A Semaphore group is used by higher layers of serial bus management to acquire the exclusive use of some sharable system resource, and subsequently to release exclusive use. It is a compound group in that its Controller provides the means to set and clear a bit that controls access to the resource, while its Flag implements the bit, possibly "in parallel with" the Flags in other Semaphore groups.

The status of the Flag represents the status of the sharable resource. When the Flag is set, the resource is allocated/assigned to the processor or process which set it. When the Flag is reset, higher layers of serial bus management can use the Controller to set the Flag and thus gain control of the resource.

6.2.4.2 Description

In Figure 6-14, page 229, the Flag part of the Semaphore group includes the HEADER RECEIVER and latch "A". The "Not-Q" output of latch "A" is called the RESOURCE FREE signal.

This Flag has a number of similarities to the Simple Flag described in Paragraph 6.1.1:

- 1) When the HEADER RECEIVER is selected by the S address in a successful frame on the serial bus, it makes its S STROBE output True, which sets latch "A" and makes RESOURCE FREE False.
- 2) When the HEADER RECEIVER is selected by the R address in a frame on the serial bus, it makes its R STROBE output True, which clears latch "A" and makes RESOURCE FREE True.
- 3) The ENABLE R input of the HEADER RECEIVER is permanently True, so that the HEADER RECEIVER never cancels a frame that has its address in the R field.

However this Flag differs from a Simple Flag in two ways:

- 1) The Flag's RESOURCE FREE output is tied back to the ENABLE S input of the Controller's HEADER RECEIVER. Because of this, if the HEADER RECEIVER is selected by the S address in a frame on the serial bus, and latch "A" is already set, the HEADER RECEIVER cancels the frame.
- 2) The CANCELLED output of the Controller's FRAME MONITOR is OR'ed with S STROBE to make the set input of latch "A" True. If another Flag also implements the semaphore bit and it is already set, it will cancel the frame. In this case, this Flag is set to bring it into compliance with the other Flag. (This feature assumes that if any module cancels a frame with the semaphore's address in the S field, a frame will be sent with the semaphore's address in the R field, to clear the Flag and allow the frame to be resent.)

REGLE 6.5:

Un indicateur qui possède un sémaphore DOIT annuler une trame ayant son adresse dans le champ S s'il est déjà positionné. Inversement, si un indicateur n'annule pas de telles trames quand il est positionné, il NE DOIT PAS être utilisé pour réaliser un sémaphore.

Le gestionnaire dans un groupe sémaphore diffère aussi des gestionnaires décrits dans les paragraphes précédents. Les couches supérieures du gestionnaire du bus série demandent le contrôle de la ressource en mettant le signal RESOURCE REQUEST à l'état vrai. Cela met le signal SEND 12 à l'état vrai si et seulement si le signal RESOURCE FREE est vrai. C'est-à-dire que l'EMETTEUR D'EN-TETE doit envoyer "une trame de positionnement" seulement si le bit du sémaphore est mis à zéro et la ressource est disponible.

RECOMMANDATION 6.2:

Pour éviter que le bus série soit surchargé par des trames de positionnement annulées inutiles, concevoir le matériel du bus série avec une liaison entre l'indicateur qui possède le bit de sémaphore et le gestionnaire qui essaie de le positionner, de telle manière que le gestionnaire n'envoie pas une trame de positionnement si l'indicateur est déjà positionné.

RECOMMANDATION 6.3:

SI le matériel du bus série est conçu avec une liaison entre l'indicateur et le gestionnaire pour éviter des trames annulées inutiles,
ET les trames envoyées par ce gestionnaire sont annulées seulement si elles essaient de positionner un bit de sémaphore qui est déjà positionné,
ALORS inclure aussi une liaison du gestionnaire vers l'indicateur comme décrit précédemment, de sorte que l'indicateur soit positionné si un autre indicateur annule une trame de positionnement envoyée par ce gestionnaire.

AUTORISATION 6.3:

Le matériel du bus série PEUT être conçu sans aucune de ces connexions, c'est-à-dire, qu'un gestionnaire indépendant comme décrit dans la section 6.1 PEUT être utilisé pour positionner des bits de sémaphore. Le prix de cette simplicité est que le bus série peut être bloqué par des trames annulées inutiles.

La bascule "B" sur la figure 6-14, page 228, commande le signal RESOURCE GRANT. Cette bascule est positionnée quand la sortie SENT du GESTIONNAIRE DE TRAME et la sortie STROBE S du RECEPTEUR D'EN-TETE deviennent vraies toutes les deux, c'est-à-dire quand le bit de sémaphore a été positionné *par ce gestionnaire*. Le signal RESOURCE GRANT indique aux couches supérieures du gestionnaire du bus série quand elles peuvent utiliser la ressource contrôlée.

RULE 6.5:

A Flag that implements a Semaphore MUST cancel a frame having its address in the S field, if it is already set. Conversely, if a Flag does not cancel such frames when it is set, it MUST NOT be used to implement a Semaphore.

The Controller in a Semaphore group also differs from Controllers described previous paragraphs. Higher layers of serial bus management request control of the resource by making RESOURCE REQUEST True. This makes SEND 12 True if and only if the RESOURCE FREE signal is True. That is, the HEADER SENDER is commanded to send a "set frame" only if the semaphore bit is cleared and the resource is available.

RECOMMENDATION 6.2:

To help prevent the serial bus from being overloaded by useless cancelled set frames, design serial bus hardware with a connection from the Flag that implements a semaphore bit to the Controller that tries to set it, so that the Controller does not send a set frame if the Flag is already set.

RECOMMENDATION 6.3:

IF serial bus hardware is designed with a connection from the Flag to the Controller to prevent useless cancelled frames,

AND frames sent by this Controller are only cancelled if they try to set a semaphore bit that is already set,

THEN also include a connection from the Controller to the Flag as described previously, so that the Flag is set if another Flag cancels a set frame sent by this Controller.

PERMISSION 6.3:

Serial bus hardware MAY be designed without either of these connections, that is, an independent Controller as described in Section 6.1 MAY be used to set semaphore bits. The price of this simplicity is that the serial bus may become clogged with useless cancelled frames.

Latch "B" in Figure 6-14, page 229, controls the RESOURCE GRANT signal. This latch is set when the SENT output of the FRAME MONITOR and the S STROBE output of the HEADER RECEIVER both go True, namely when the semaphore bit has been set by *this Controller*. RESOURCE GRANT signals higher layers of serial bus management when they can use the controlled resource.

Les couches supérieures du gestionnaire du bus série maintiennent le signal de RESOURCE REQUEST à l'état vrai jusqu'à ce que le signal RESOURCE GRANT devienne vrai et jusqu'à ce qu'elles n'aient plus besoin de la ressource. Puis, quand elles mettent le signal RESOURCE REQUEST à l'état faux, cela entraîne le passage du signal SEND 21 du gestionnaire à l'état vrai. Ce qui entraîne qu'il envoie une trame avec l'adresse du sémaphore dans le champ R. Cela met à zéro l'indicateur ou les indicateurs de sémaphore ainsi que la bascule de verrouillage RESOURCE GRANT qui se trouve sur la carte.

OBSERVATION 6.3:

Différents systèmes de bus série peuvent avoir des approches différentes du nombre de sémaphores qui peuvent être commandés sur le bus série. Un système peut allouer un grand nombre d'adresses pour les ressources partagées, tandis qu'un autre peut utiliser une adresse contrôlant le droit d'accès à une table en mémoire des ressources partagées, chaque ressource ayant un "bit d'occupation" dans la table.

OBSERVATION 6.4:

Différents systèmes de bus série peuvent avoir des approches différentes pour affecter le matériel du bus série aux sémaphores. S'il y a un petit nombre de sémaphores (par exemple un seul), il est possible d'affecter un indicateur sur chaque carte pour surveiller l'état du sémaphore, et probablement même un gestionnaire sur chaque carte, pour le positionnement et la remise à zéro du sémaphore. Cependant, dans les petits systèmes, il peut y avoir seulement un groupe gestionnaire *par carte*, utilisé pour tous les objectifs y compris les sémaphores.

OBSERVATION 6.5:

La description fonctionnelle suivante est écrite de la manière la moins présomptive, c'est-à-dire en supposant que ni les indicateurs ni les gestionnaires ne sont dédiés à long terme à l'usage des sémaphores.

6.2.4.3 Fonctionnement

Quand un système de bus série est configuré pour la première fois, deux types d'adresses spéciales sont assignées, en plus de celles utilisées pour sélectionner les indicateurs simples, les parleurs et les écouteurs.

- 1) Adresses de sémaphore: une adresse est assignée à chaque ressource partageable qui peut être contrôlée par des sémaphores du bus série.
- 2) Codes des demandeurs: une adresse est assignée à chaque groupe de sémaphores ou entité des couches supérieures qui peut avoir besoin de demander le contrôle d'une ressource par les moyens des sémaphores du bus série.

REGLE 6.6:

Si deux groupes de sémaphores sont utilisés pour tenter d'acquérir la même ressource partagée, ils DOIVENT utiliser des codes de demandeur différents.

Higher layers of serial bus management retain RESOURCE REQUEST True until RESOURCE GRANT becomes True, and continue to do so until they have no further need for the resource. Then, when they make RESOURCE REQUEST False, this makes the SEND 21 signal to the Controller True, causing it to send a frame with the semaphore's address in the R field. This resets (all the) the semaphore Flag(s) and the RESOURCE GRANT latch on this board as well.

OBSERVATION 6.3:

Different serial bus systems will have differing approaches to the number of semaphores that can be controlled on the serial bus. One system might allocate a large number of addresses for shared resources, while another might use one address controlling the right to access a memory table of shared resources, each resource having a "busy bit" in the table.

OBSERVATION 6.4:

Different serial bus systems will have differing approaches to dedicating serial bus hardware to semaphores. If there is a small number of semaphores (e.g. one), it may be feasible to dedicate a Flag on each board to tracking the semaphore status, and possibly even a Controller on each board, for semaphore setting and clearing. However, in smaller systems there may be just one Controller group *per board*, which is used for all serial bus purposes including semaphores.

OBSERVATION 6.5:

The following operational description is written in the least-presumptive way, that is, assuming that neither Flags nor Controllers are dedicated long-term to semaphore use.

6.2.4.3 Operation

When a serial bus system is first configured, two types of special addresses are assigned in addition to those used to select Simple Flags, Talkers and Listeners:

- 1) Semaphore addresses: one address is assigned to each sharable resource that can be controlled by serial bus semaphores.
- 2) Requester codes: one address is assigned for each Semaphore group or higher-layer entity that may need to request control of a resource by means of serial bus semaphores.

RULE 6.6:

If two Semaphore groups are used to try to acquire the same shared resource, they **MUST** use different requester codes.

SUGGESTION 6.4:

Pendant l'allocation d'adresse, positionner les codes de demandeurs de l'une des deux manières suivantes:

- 1) Avec une adresse pour chaque groupe sémaphore, c'est-à-dire chaque gestionnaire qui pourrait envoyer un en-tête pour positionner un sémaphore. Le groupe sémaphore utiliserait alors le même code de demandeur chaque fois qu'il voudrait acquérir une ressource partagée.
- 2) Avec une adresse pour chaque entité des couches supérieures (c'est-à-dire processus ou tâche) qui pourrait essayer de positionner un sémaphore. Cela garantit un code de demandeur unique en supposant qu'une entité donnée peut seulement demander le positionnement au moyen d'un seul groupe de sémaphores à la fois.

Puis, quand les couches supérieures du gestionnaire du bus série ont besoin d'acquérir une ressource partagée du système contrôlée par un sémaphore du bus série, elles configurent un groupe sémaphore comme suit:

- 1) Elles positionnent le port de priorité de l'EMETTEUR D'EN-TETE à une valeur appropriée.
- 2) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE à l'adresse de sémaphore de la ressource demandée.
- 3) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE au code du demandeur.
- 4) Elles mettent l'entrée ENABLE R du RECEPTEUR D'EN-TETE à l'état vrai.
- 5) Si besoin est, elles remettent à zéro l'indicateur sur la carte afin que l'entrée ENABLE S du RECEPTEUR D'EN-TETE soit vraie.
- 6) Elles mettent le signal de RESSOURCE REQUEST à l'état vrai. Cela met l'entrée SEND 12 du RECEPTEUR D'EN-TETE à l'état vrai.

OBSERVATION 6.6:

Quand un groupe sémaphore est configuré comme décrit ci-dessus, son indicateur peut être réuni à plusieurs autres indicateurs qui répondent à l'adresse de la ressource partagée. Evidemment, dans ce cas, il n'y a pas de manière de prédire si la ressource est utilisée ou non, c'est-à-dire si l'indicateur est positionné ou remis à zéro. On suppose que l'indicateur est remis à zéro, si d'autres indicateurs utilisant le bit de sémaphore sont positionnés, cet indicateur s'alignera sitôt que la divergence aura été signalée.

L'EMETTEUR D'EN-TETE commence alors à essayer d'envoyer un en-tête ayant l'adresse de sémaphore dans le champ S et le code du demandeur comme adresse R. D'abord, il attend (si besoin est) que le MODULE DE SURVEILLANCE DE TRAME signale qu'aucune trame n'est en cours sur le bus série. Puis il envoie un bit_départ et commence son en-tête. Le gestionnaire peut obtenir les résultats suivants dans l'attente de la transmission:

SUGGESTION 6.4:

During address allocation, set aside requester codes in one of two ways:

- 1) With one address for each Semaphore group, that is, each Controller that might send a Header to set a semaphore. The Semaphore group would then use the same requester code any time it was used to acquire any shared resource.
- 2) With one address for each higher-layer entity (e.g. a process or task) that might try to set a semaphore. This guarantees requester code uniqueness assuming that a given entity can only request the setting of a semaphore via one Semaphore group at a time.

Then, when higher layers of serial bus management need to acquire a shared system resource that is controlled by a serial bus semaphore, they configure a Semaphore group as follows:

- 1) They set the Priority Port of the HEADER SENDER to some appropriate value.
- 2) They set the Code 1 Port of the HEADER SENDER and the Code Port of the HEADER RECEIVER to the semaphore address for the needed resource.
- 3) They set the Code 2 Port of the HEADER SENDER to the requester code.
- 4) They make the ENABLE R input of the HEADER RECEIVER True.
- 5) If necessary, they reset the on-board Flag, so that the ENABLE S input of the HEADER RECEIVER is True.
- 6) They make the RESOURCE REQUEST signal True. This makes the SEND 12 input of the HEADER SENDER True.

OBSERVATION 6.6:

When a Semaphore group is configured as described above, its Flag may be joining a number of other Flags that respond to the address of the shared resource. Of course, in this case there is no way to predict whether the resource is in use or not, i.e. whether the Flag should be set or cleared. The assumption is made that the Flag is cleared; if other Flags implementing the semaphore bit are set, this Flag will be made to conform as soon the discrepancy is signalled.

The HEADER SENDER then begins to try to send a Header with the semaphore's address in the S field and the requester code as the R address. First, it waits (if necessary) for its FRAME MONITOR to signal that no serial bus frame is in progress. Then it sends a Start_bit and begins its Header. The Controller may encounter the following results in the transmission attempt:

- 1) Il peut obtenir le bus série et positionner l'indicateur. C'est le cas typique. S'il y a d'autres indicateurs (dans d'autres groupes sémaphore) configurés à la même adresse, ils sont aussi positionnés.
- 2) Il peut perdre le bus série au profit d'une trame non reliée à un sémaphore ou qui remet à zéro le sémaphore. Dans l'un ou l'autre de ces cas, il peut simplement essayer à nouveau d'envoyer l'en-tête et positionner le sémaphore.
- 3) Il peut perdre le bus série au profit d'un autre en-tête qui positionne le sémaphore. Dans ce cas, cet indicateur de groupe est aussi positionné, ce qui entraîne que le groupe diffère l'envoi de la "trame de positionnement" à nouveau, jusqu'à ce que l'indicateur soit remis à zéro.
- 4) Il peut gagner le bus et avoir sa trame annulée. Cela signifie que l'indicateur est positionné dans un autre groupe sémaphore et que la ressource partagée est utilisée. Dans ce cas, la sortie annulation du MODULE DE SURVEILLANCE DE TRAME positionne cet indicateur du groupe sémaphore pour se mettre en conformité avec le ou les autres indicateurs. Comme dans les cas précédents, l'état positionné de l'indicateur entraîne que le groupe diffère l'envoi de la "trame de positionnement" à nouveau, jusqu'à ce que l'indicateur soit remis à zéro.
- 5) Il peut perdre le bus série au profit d'une trame qui essaie de positionner le sémaphore, mais voit sa trame annulée. Dans ce cas, il essaie à nouveau d'envoyer l'en-tête et se trouve probablement dans le cas 3) la fois suivante.

Puisque différents groupes sémaphores utilisent différents codes de demandeur, quand un gestionnaire réussit dans l'envoi de sa "trame de positionnement", l'arbitrage du bus série garantit que c'est seulement lui qui a obtenu le bus et qui positionne le sémaphore. Les sorties SENT du gestionnaire positionnent le signal RESOURCE GRANT et ainsi notifient aux couches supérieures du gestionnaire du bus série que la ressource est disponible.

Quand l'utilisation d'une ressource est terminée, les couches supérieures mettent le signal RESOURCE REQUEST à l'état faux. Cela met SEND 21 à l'état vrai et entraîne que le gestionnaire envoie un en-tête avec le code de demandeur comme adresse S et l'adresse de sémaphore dans le champ R. Cette trame remet à zéro (toutes les copies de) l'indicateur et remet aussi à zéro la bascule RESOURCE GRANT. Si d'autres gestionnaires attendaient que leurs copies de l'indicateur soient remises à zéro, ils seraient libérés pour entrer en compétition pour le sémaphore.

AUTORISATION 6.4:

Un en-tête dont le but est de remettre à zéro un sémaphore PEUT avoir soit "tout-à-un" dans son adresse S, ou le même code de demandeur utilisé dans l'adresse R de la trame qui le positionne.

- 1) It may win the serial bus arbitration and set the Flag. This is the typical case. If there are other Flags (in other Semaphore groups) configured for the same address, they are set also.
- 2) It may lose the serial bus arbitration to a frame not related to the semaphore, or to one that clears the semaphore. In either of these cases, it merely retries the attempt to send the Header and set the semaphore.
- 3) It may lose the serial bus arbitration to another Header, which sets the semaphore. In this case, this group's Flag is set also, which causes the group to delay sending the "set frame" again, until the Flag is cleared.
- 4) It may win the serial bus arbitration and have the frame cancelled. This means that the Flag is set in another Semaphore group, and that the shared resource is in use. In this case, the Cancelled output of the FRAME MONITOR sets this Semaphore group's Flag to bring it into conformance with the other Flag(s). As in the previous case, the set state of the Flag makes the group delay sending the "set frame" again, until the Flag is cleared.
- 5) It may lose the serial bus arbitration to a frame that tries to set the semaphore, but has the frame cancelled. In this case, it retries sending the Header, probably leading to case 3 next time.

Since different Semaphore groups use different Requester codes, when a Controller succeeds in sending its "set frame", the serial bus arbitration has ensured that it is the only one that won the arbitration and set the Semaphore. The Controller's SENT output sets the RESOURCE GRANT signal and thus notifies higher layers of serial bus management that the resource is available for use.

When the use of the resource is complete, the higher layers make RESOURCE REQUEST False. This makes SEND 21 True and causes the Controller to send a Header with the requester code as the S address and the semaphore's address in the R field. This frame clears (all copies of) the Flag, and also clears the RESOURCE GRANT latch. If other Controllers have been waiting for their copies of the Flag to be cleared, they are released to contend for the semaphore.

PERMISSION 6.4:

A Header intended to clear a Semaphore MAY have either "all-ones" as its S address, or the same Requester code used in the R address of the frame that set it.

6.2.5 Sémaphore à vérification de signature

Un sémaphore à vérification de signature est représenté par la figure 6-15 ci-dessous. Il diffère du groupe sémaphore décrit dans le paragraphe 6.2.4 en cela que, lorsqu'il est sélectionné par une adresse S ou R dans un en-tête, il examine aussi l'autre adresse pour déterminer sa réponse à la trame.

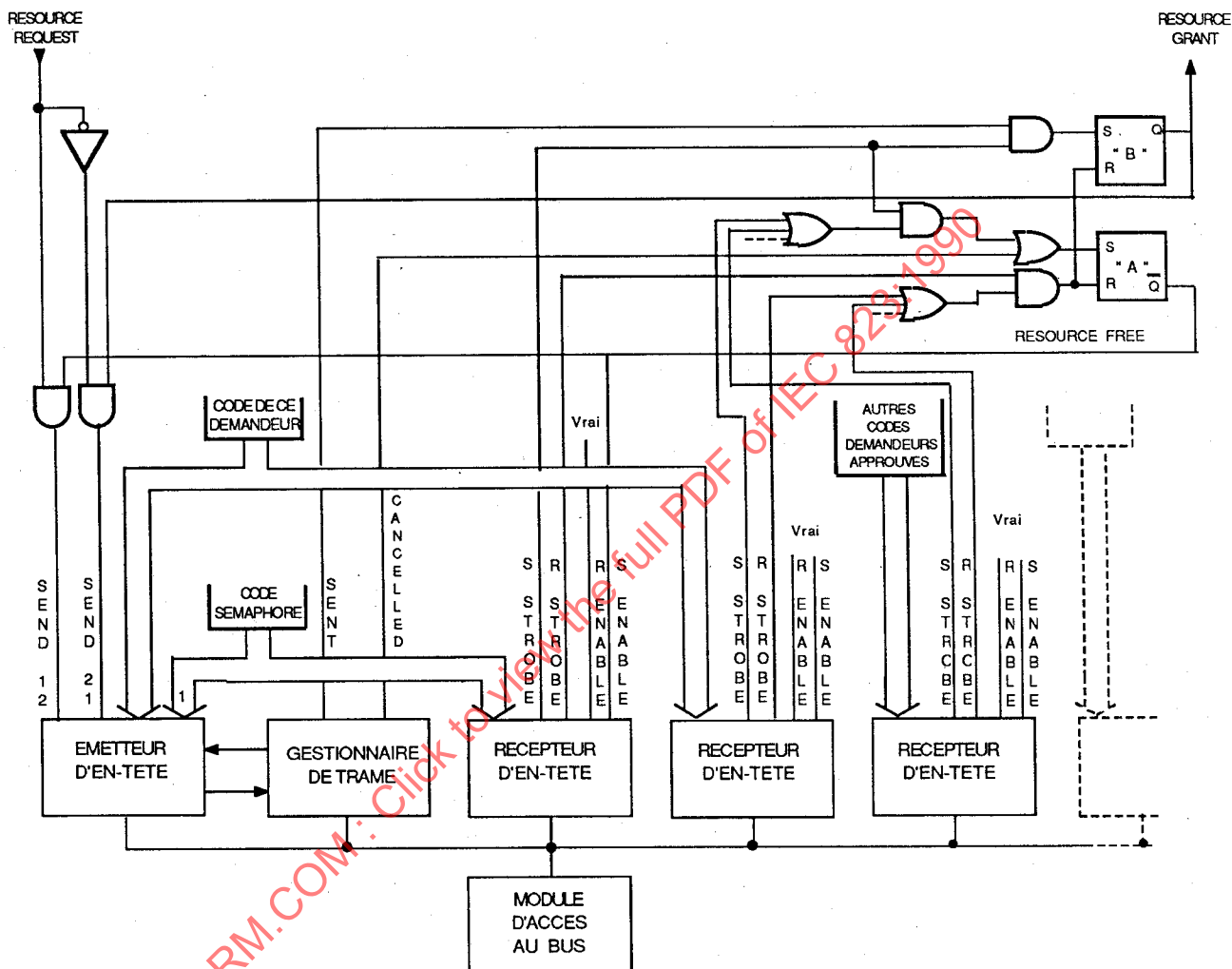


Figure 6-15 - Sémaphore à vérification de signature

AUTORISATION 6.5:

Quand un groupe quelconque de modules du bus série est sélectionné par l'adresse S dans un en-tête, il PEUT aussi utiliser l'adresse R pour déterminer s'il répond, ou comment il répond.

AUTORISATION 6.6:

Quand un groupe quelconque de modules du bus série est sélectionné par l'adresse R dans un en-tête, il PEUT aussi utiliser l'adresse S pour déterminer s'il répond, ou comment il répond.

6.2.5 Signature-Checking Semaphore

A Signature-Checking Semaphore is shown in Figure 6-15. It differs from the Semaphore group described in Paragraph 6.2.4 in that, when it is selected by the S or R address in a Header, it also examines the other address to determine its response to the frame.

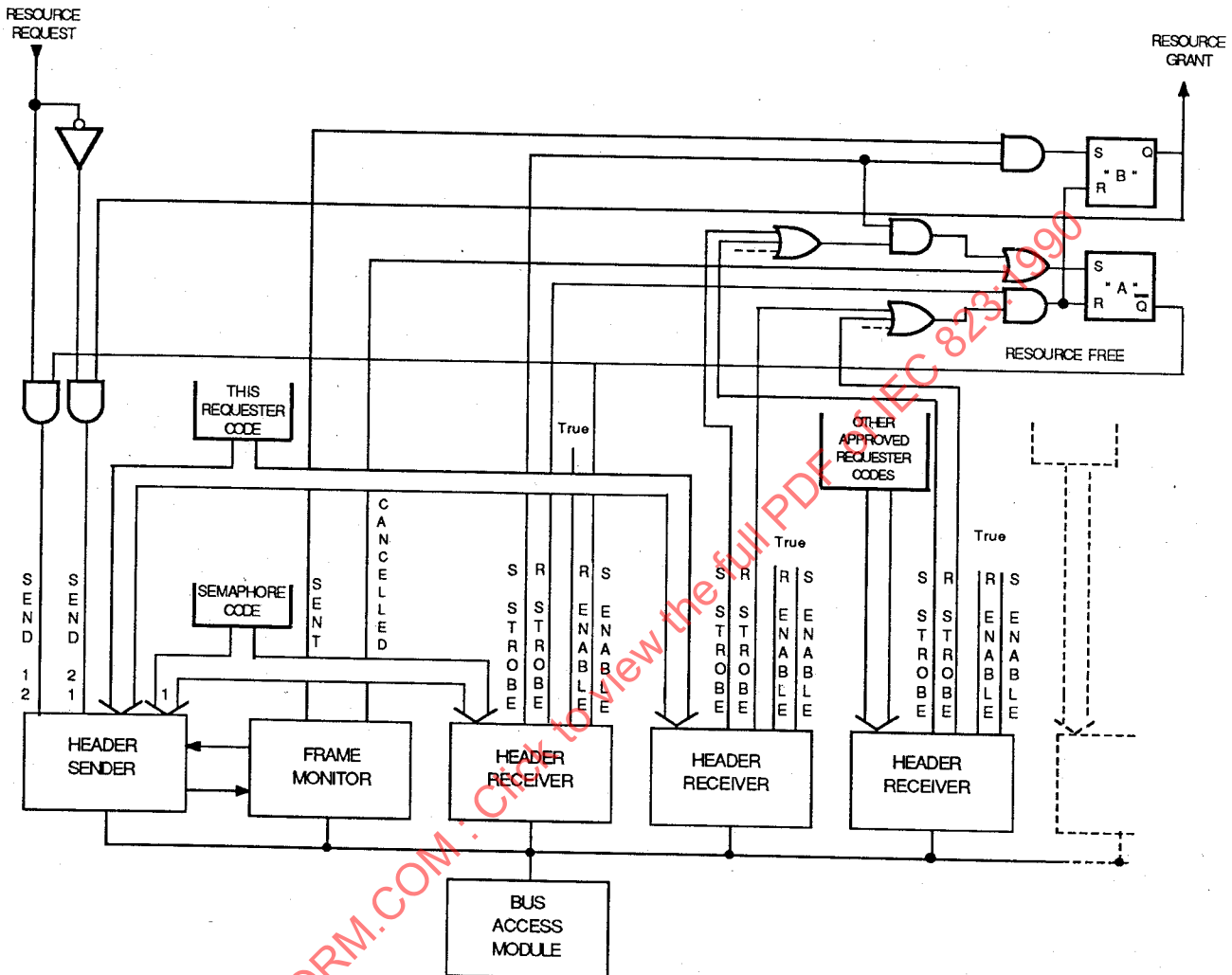


Figure 6-15 - Signature-Checking Semaphore

PERMISSION 6.5:

When any serial bus module group is selected by the S address in a Header, it MAY also use the R address to determine whether it responds, or how it responds.

PERMISSION 6.6:

When any serial bus module group is selected by the R address in a Header, it MAY also use the S address to determine whether it responds, or how it responds.

6.2.5.1 But

Ce groupe a exactement le même but que le sémaphore décrit précédemment, excepté qu'il ajoute un mécanisme par lequel le sémaphore peut seulement être atteint par des trames avec certains codes de demandeur (ceux qui ont été identifiés comme ayant le droit de le faire). Cette caractéristique peut être demandée dans certaines applications de haute sécurité.

6.2.5.2 Description

Le sémaphore à vérification de signature montré dans la figure 6.15, page 240, consiste en un gestionnaire et un indicateur comme le groupe sémaphore, excepté que son indicateur comprend plusieurs RECEPTEURS D'EN-TETE supplémentaires, un pour chacun des codes de demandeur qui sont autorisés à accéder à l'indicateur et changer l'état du bit de sémaphore. Un de ces modules supplémentaires (appelé RECEPTEUR D'EN-TETE de demandeur) pourrait avoir le même code de demandeur que celui présenté sur le port de code 2 de l'EMETTEUR D'EN-TETE.

Le RECEPTEUR D'EN-TETE de demandeur assure que les deux adresses S et R sont décodées par chaque trame sur le bus série, et que le sémaphore répond seulement à celles contenant l'adresse de sémaphore correcte ET un code de demandeur "approuvé". Les sorties R STROBE des RECEPTEURS D'EN-TETE de demandeur sont reliées en OU logique et le R STROBE résultant est relié en ET avec le S STROBE du RECEPTEUR D'EN-TETE qui surveille l'adresse de sémaphore pour élaborer le signal qui positionne l'indicateur (bascule "A"). De même, les sorties S STROBE des RECEPTEURS D'EN-TETE de demandeur sont reliées en OU logique et le S STROBE résultant est relié en ET avec le R STROBE du RECEPTEUR D'EN-TETE qui surveille l'adresse de sémaphore pour élaborer le signal qui remet à zéro l'indicateur et le signal RESOURCE GRANT (bascules "A" et "B").

OBSERVATION 6.7:

Cette fonction d'examen de "l'autre" adresse dans les trames du bus série est formellement spécifiée ici comme un tableau de RECEPTEUR D'EN-TETE. En pratique, la fonction pourrait être appliquée par des structures de 1024 x 1 bit de RAM indexée par l'adresse ou par un ensemble de 1, 2 ou plus de comparateurs d'amplitude.

6.2.5.3 Fonctionnement

Quand un système de bus série utilisant des sémaphores à vérification de signature est configuré, les sémaphores et codes de demandeur sont alloués comme décrit précédemment pour les groupes sémaphore. De plus, un ensemble de privilèges d'accès sont déterminés, indiquant quels codes de demandeur peuvent être utilisés avec chaque adresse de sémaphore.

Puis, lorsque les couches supérieures du gestionnaire du bus série ont besoin d'acquérir une ressource partagée du système qui est contrôlée par un sémaphore à vérification de signature, elles configurent le groupe de la manière suivante:

6.2.5.1 Purpose

This group has exactly the same purpose as the Semaphore described previously, except that it adds a mechanism by which the semaphore can only be accessed by frames with certain requester codes (those that have been identified as having the right to do so). This feature may be required in certain high-security applications.

6.2.5.2 Description

The Signature-Checking Semaphore shown in Figure 6-15, page 241, consists of a Controller and a Flag as does the Semaphore group, except that its Flag includes a number of additional HEADER RECEIVERS, one for each of the Requester codes that are allowed to access the Flag and change the status of the Semaphore bit. One of the additional modules (called requester HEADER RECEIVERS) should have the same Requester code that is presented at the Code 2 port of the HEADER SENDER.

The requester HEADER RECEIVERS ensure that both the S and R addresses are decoded for each frame on the serial bus, and that the semaphore responds only to those containing the proper semaphore address AND an "approved" requester code. The R STROBE outputs of the requester HEADER RECEIVERS are OR'ed, and this composite R STROBE is AND'ed with the S STROBE of the HEADER RECEIVER that monitors the Semaphore address, to make the signal that sets the Flag (latch "A"). Similarly, the S STROBE outputs of the requester HEADER RECEIVERS are OR'ed, and this composite S STROBE is AND'ed with the R STROBE of the HEADER RECEIVER that monitors the Semaphore address, to make the signals that clears the Flag and RESOURCE GRANT (latches "A" and "B").

OBSERVATION 6.7:

This function of examining the "other" address in serial bus frames is formally specified here as an array of HEADER RECEIVERS. In practice, the function might be implemented by such structures as a 1024 x 1 bit RAM indexed by the address, or by a set of 1, 2 or more magnitude comparators.

6.2.5.3 Operation

When a serial bus system using Signature-Checking Semaphores is configured, Semaphore and Requester codes are allocated as described earlier for Semaphore groups. In addition, a set of access privileges are determined, indicating which requester codes can be used with each semaphore address.

Then, when higher layers of serial bus management need to acquire a shared system resource that is controlled by a Signature-Checking Semaphore, they configure the group as follows:

- 1) Elles positionnent le port de priorité de l'EMETTEUR D'EN-TETE à une valeur appropriée.
- 2) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE de sémaphore à l'adresse du sémaphore concernant la ressource requise.
- 3) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE et le port de code d'un des RECEPTEURS D'EN-TETE de demandeur selon le code du demandeur.
- 4) Elles positionnent les ports de code des autres RECEPTEURS D'EN-TETE demandeurs selon les autres codes de demandeurs qui sont autorisés à accéder au bit de sémaphore concernant de la ressource requise.
- 5) Elles mettent les entrées ENABLE R de tous les RECEPTEURS D'EN-TETE à l'état vrai.
- 6) Elles mettent les entrées ENABLE S des RECEPTEURS D'EN-TETE de demandeur à l'état vrai.
- 7) Si nécessaire, elles réinitialisent l'indicateur de la carte afin que l'entrée ENABLE S du RECEPTEUR D'EN-TETE du sémaphore soit à l'état vrai.
- 8) Elles mettent le signal RESOURCE REQUEST à l'état vrai. Cela met l'entrée SEND 12 de l'EMETTEUR D'EN-TETE à l'état vrai.

Le fonctionnement se poursuit alors comme il a été décrit auparavant pour les groupes de sémaphores, tant que le code du demandeur utilisé pour accéder au bit de sémaphore est l'un de ceux qui ont été autorisés pour ce faire. Dans le cas contraire, le RECEPTEUR D'EN-TETE du sémaphore produira l'impulsion d'échantillonnage attendue, mais aucun des RECEPTEURS D'EN-TETE de demandeur ne produira l'impulsion complémentaire et l'indicateur du sémaphore ne sera pas modifié.

OBSERVATION 6.8:

Avec des groupes de sémaphores comme il a été décrit auparavant, les contenus de la sous-trame d'état de trame sont 010 pour une "trame réussie de mise à un" et 001 pour une "trame réussie de mise à zéro". Avec des sémaphores à contrôle de signature, il convient que l'état de trame contienne 011 pour chacun des types de trame.

REGLE 6.7:

Un en-tête destiné à mettre à zéro un sémaphore à contrôle de signature DOIT posséder dans son adresse S le même code de demandeur que celui utilisé dans l'adresse R de la trame qui l'a mis à un.

- 1) They set the Priority Port of the HEADER SENDER to some appropriate value.
- 2) They set the Code 1 Port of the HEADER SENDER and the Code Port of the semaphore HEADER RECEIVER to the semaphore address for the needed resource.
- 3) They set the Code 2 Port of the HEADER SENDER and the Code Port of one of the requester HEADER RECEIVERS to the requester code.
- 4) They set the Code Ports of the other requester HEADER RECEIVERS to the other requester codes that are allowed to access the semaphore bit for the needed resource.
- 5) They make the ENABLE R inputs of all the HEADER RECEIVERS True.
- 6) They make the ENABLE S inputs of the requester HEADER RECEIVERS True.
- 7) If necessary, they reset the on-board Flag, so that the ENABLE S input of the semaphore HEADER RECEIVER is True.
- 8) They make the RESOURCE REQUEST signal True. This makes the SEND 12 input of the HEADER SENDER True.

Operation then proceeds as described earlier for Semaphore groups, as long as the requester code used to access the semaphore bit is one of those allowed to do so. If not, the semaphore HEADER RECEIVER will produce the expected strobe, but none of the requester HEADER RECEIVERS will produce the complementary strobe, and the semaphore Flag will not be affected.

OBSERVATION 6.8:

With Semaphore groups, as described earlier, the contents of the Frame Status subframe are 010 for a "successful set frame", and 001 for a "successful reset frame". With Signature-Checking Semaphores, the Frame Status should contain 011 for either type of frame.

RULE 6.7:

A Header intended to clear a Signature-Checking Semaphore MUST have the same Requester code in its S address as was used in the R address of the frame that set it.

6.2.6 Groupe à passage de jeton

Un groupe à passage de jeton est représenté par la figure 6-16 ci-dessous. Il est composé d'un gestionnaire, d'un indicateur, et d'autres circuits logiques, disposés de manière à fournir un mécanisme à passage de jeton qui nécessite seulement une attention minimale des couches supérieures du gestionnaire du bus série.

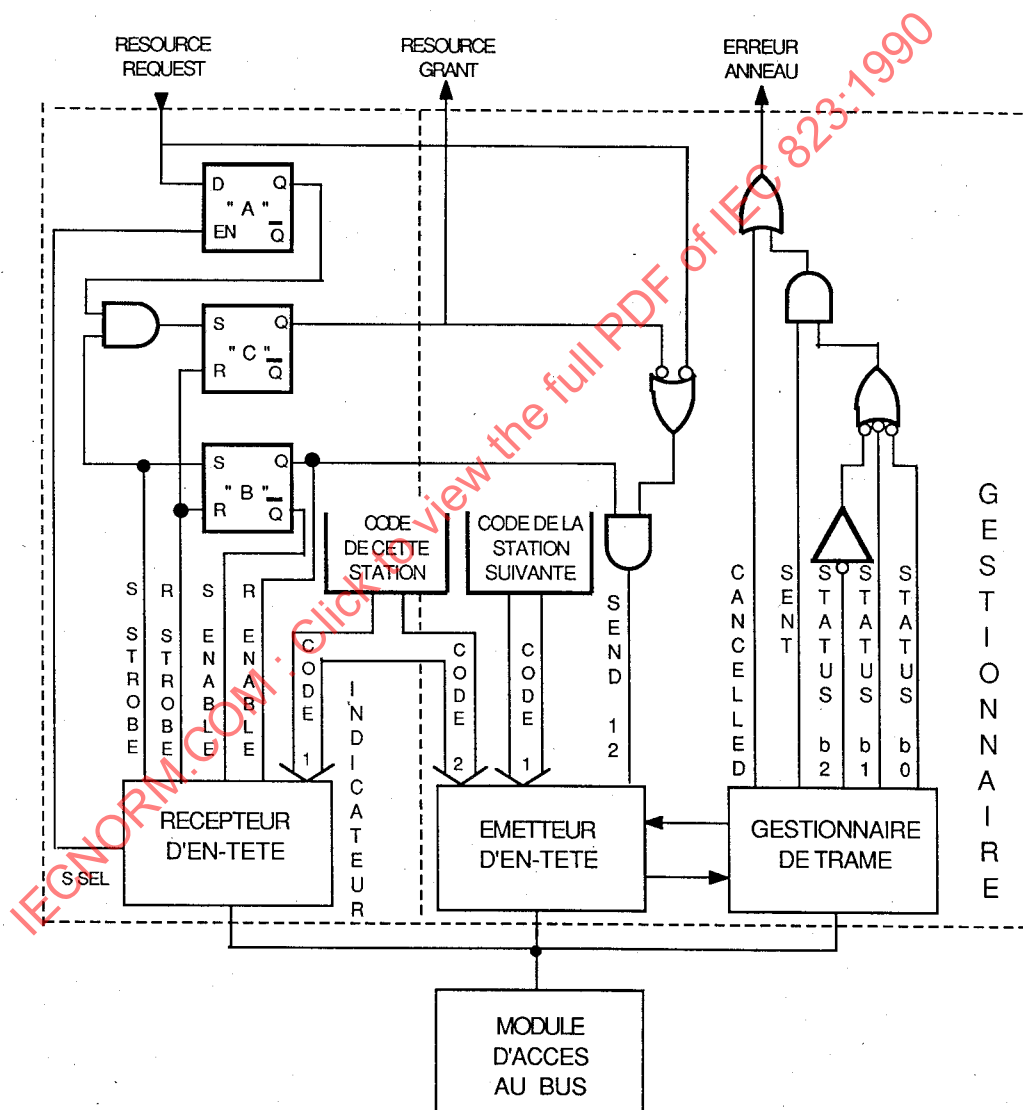


Figure 6-16 - Groupe à passage de jeton

6.2.6 Token Passing Group

A Token Passing Group is shown in Figure 6-16. It is composed of a Controller, a Flag and other logic, arranged to provide a token passing mechanism that requires only minimal attention from higher layers of serial bus management.

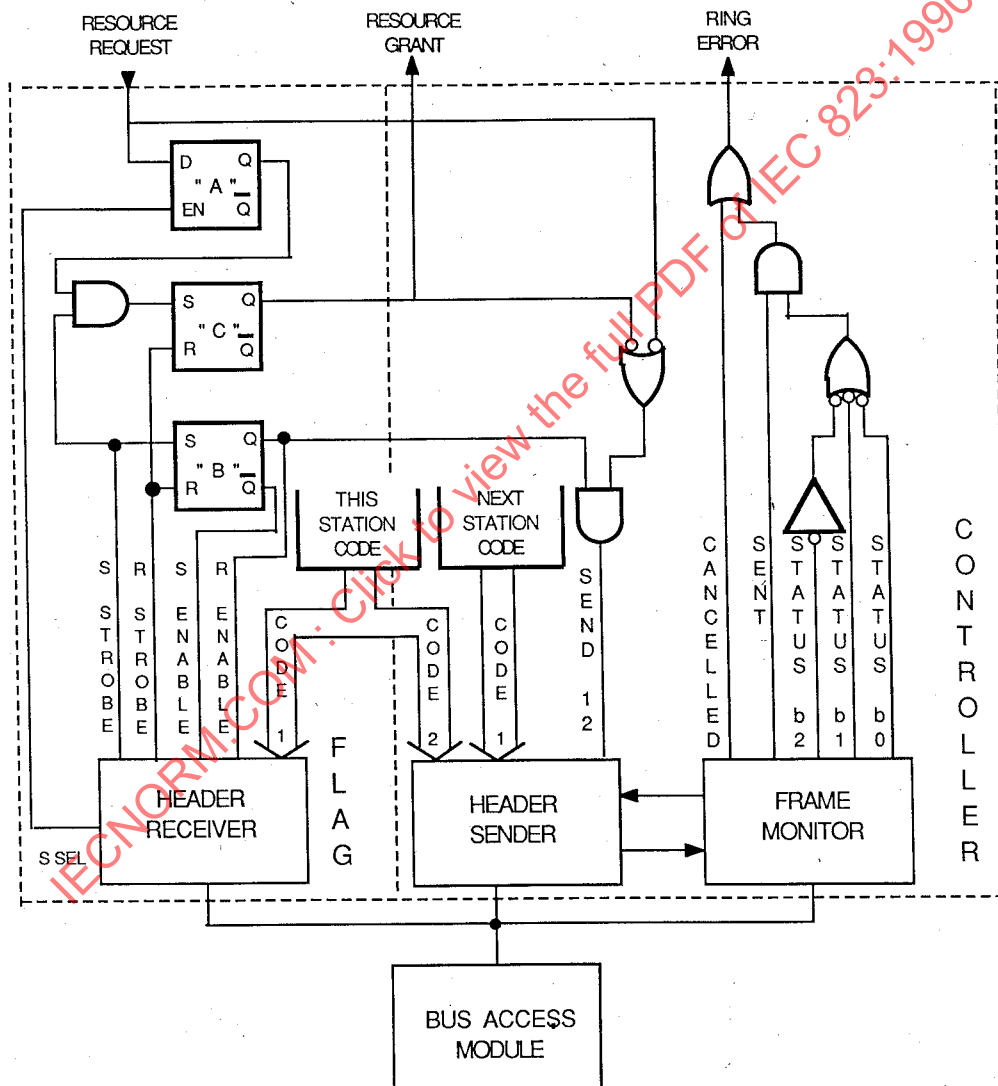


Figure 6-16 - Token Passing Group

6.2.6.1 But

Un ensemble de groupes à passage de jeton peut être utilisé dans un système pour allouer/affecter une ressource partageable d'une autre façon que par l'usage de sémaphores. Chaque "utilisateur" de la ressource possède un groupe à passage de jeton qui lui est dédié et auquel on affecte une adresse. Ces groupes sont configurés logiquement en "anneau" de manière que chacun possède un groupe "successeur" particulier, dont il connaît l'adresse. Les groupes à passage de jeton fonctionnent en émettant des trames, chacun vers son successeur. Un "utilisateur" ne peut utiliser la ressource qui est gérée que seulement après que son groupe a reçu une trame, et avant que son groupe n'ait émis une trame vers son successeur.

6.2.6.2 Fonctionnement normal

Lorsque l'anneau à passage de jeton est établi, les couches supérieures du gestionnaire du bus série configurent un groupe à passage de jeton de la manière suivante:

- 1) Elles positionnent le port de priorité de l'EMETTEUR D'EN-TETE à une valeur appropriée.

RECOMMANDATION 6.4:

Puisqu'un anneau à passage de jeton génère un niveau de trafic élevé sur le bus série, positionner la priorité de tous ses groupes à une faible valeur (par exemple 0) pour permettre aux autres trames du bus série de passer avec des interférences réduites au minimum.

- 2) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE à l'adresse de ce groupe.
- 3) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE à l'adresse du successeur de ce groupe.
- 4) Elles mettent RESOURCE REQUEST à l'état faux.
- 5) Elles remettent à zéro les bascules à verrouillage A, B, et C de tous les groupes à passage de jeton, sauf la bascule B de celui désigné pour débiter le fonctionnement de l'anneau. Dans ce groupe, elles mettent à un la bascule B, ce qui positionne SEND 12 de l'EMETTEUR D'EN-TETE de ce groupe à l'état vrai afin qu'il émette la première trame.

Ensuite, lorsque les couches supérieures du gestionnaire du bus série ont besoin d'acquiescer la ressource qui est gérée, elles positionnent le signal RESOURCE REQUEST à l'état vrai et attendent que le signal RESOURCE GRANT passe à l'état vrai.

Lorsqu'un groupe prédécesseur émet un en-tête, le RECEPTEUR D'EN-TETE est sélectionné par l'adresse S; il envoie une impulsion à l'état vrai sur S SELECT. Cela échantillonne l'état courant du signal RESOURCE REQUEST dans la bascule A transparente. En supposant

6.2.6.1 Purpose

A set of Token Passing Groups can be used in a system to allocate/assign a sharable resource in an alternative way to the use of semaphores. Each "user" of the resource has a dedicated Token Passing Group which is assigned an address. These groups are logically configured in a "ring", such that each has a particular "successor" group, for which it knows the address. The Token Passing Groups operate by sending frames, each to its successor. A "user" can only utilize the controlled resource after its group has been sent a frame, and before its group has sent a frame to its successor.

6.2.6.2 Normal Operation

When the token passing ring is established, higher layers of serial bus management configure a Token Passing Group as follows:

- 1) They set the Priority Port of the HEADER SENDER to some appropriate value.

RECOMMENDATION 6.4:

Since a token passing ring generates a high level of serial bus traffic, set the Priority of all its groups at a low value (e.g. 0) to allow other serial bus frames to get through with minimal interference.

- 2) They set the Code 2 Port of the HEADER SENDER and the Code Port of the HEADER RECEIVER to this group's address.
- 3) They set the Code 1 Port of the HEADER SENDER to the address of this group's successor.
- 4) They make RESOURCE REQUEST False.
- 5) They clear latches A, B and C at all Token Passing Groups, except for latch B at the one designated to begin ring operation. At that group, they set latch B, making SEND 12 to that group's HEADER SENDER True, so that it sends the first frame.

Then, when higher layers of serial bus management need to acquire the controlled resource, they make the RESOURCE REQUEST signal True and wait for the RESOURCE GRANT signal to go True.

When a predecessor group sends a Header, the HEADER RECEIVER is selected by the S address; it pulses S SELECT True. This samples the current state of the RESOURCE REQUEST signal into transparent latch A. Assuming that latch B is cleared, S ENABLE is True and the

que la bascule B soit mise à zéro, S ENABLE est à l'état vrai et le RECEPTEUR D'EN-TETE n'invalidé pas la trame. Supposant aussi qu'aucun autre module du bus série n'invalidé cette trame, le RECEPTEUR D'EN-TETE émet une impulsion sur S STROBE après avoir surveillé la trame entière, s'il la considère correcte.

Le signal S STROBE met à un la bascule B. Si la bascule A a été remise à zéro auparavant du fait que RESOURCE REQUEST était à l'état faux, cela met immédiatement SEND 12 à l'état vrai dans l'EMETTEUR D'EN-TETE, qui émet un en-tête pour passer la trame de jeton à son groupe successeur.

Cependant, si la bascule A a été mise à un auparavant du fait que RESOURCE REQUEST était à l'état vrai, S STROBE ne met pas immédiatement SEND 12 à l'état vrai, mais met à un la bascule C. Cela met RESOURCE GRANT à l'état vrai, signalant aux couches supérieures de la logique du bus série que la ressource gérée est disponible pour son usage.

Lorsque les couches supérieures ont fini d'utiliser la ressource, elles positionnent RESOURCE REQUEST à l'état faux, ce qui rend SEND 12 vrai à l'EMETTEUR D'EN-TETE afin qu'il émette un en-tête pour passer le jeton au groupe suivant.

L'en-tête contient l'adresse du groupe successeur dans le champ S et l'adresse de ce groupe dans le champ R. Le RECEPTEUR D'EN-TETE du successeur détecte son adresse, positionne son indicateur et émet 010 dans le champ de mot d'état de la trame. Le RECEPTEUR D'EN-TETE de ce groupe détecte son adresse, remet à zéro son indicateur et émet 001 dans le champ de mot d'état de la trame. Ces valeurs sont combinées en OU logique par le bus série pour obtenir 011, le mot d'état correct pour une trame à passage de jeton.

Cette séquence de fonctionnement normal se poursuit automatiquement autour de l'anneau à passage de jeton sans requérir l'attention des couches supérieures, excepté lorsque c'est nécessaire pour utiliser la ressource qui est gérée.

6.2.6.3 Détection des erreurs

Les sorties de la bascule B sont reliées aux entrées S et R ENABLE du RECEPTEUR D'EN-TETE afin que l'indicateur dans un groupe à passage de jeton invalide une trame qui essaie de le positionner à un s'il est déjà positionné à cette valeur, et invalide aussi une trame qui essaie de le mettre à zéro s'il y est déjà positionné. Ces caractéristiques aident à assurer que le fonctionnement se poursuit et qu'une seule trame à jeton circule sur l'anneau. Il convient que de telles trames invalidées ne se produisent pas lors du fonctionnement normal d'un anneau à passage de jeton.

La logique reliée aux sorties du GESTIONNAIRE DE TRAME détecte de telles invalidations et vérifie également le mot d'état de trame en ce qui concerne la réponse par les RECEPTEURS D'EN-TETE à la fois dans son groupe et dans le groupe successeur. Au cas où il y a un problème quelconque, une impulsion à l'état vrai de la sortie RING ERROR est émise vers les couches supérieures du gestionnaire du bus série.

RECOMMANDATION 6.5:

En plus de la vérification d'erreur permise par la logique traitant RING ERROR, il convient que les couches supérieures gèrent un chien de garde qui est déclenché lorsqu'une trame à jeton est reçue, et dont l'intervalle temporel est de:

(NOMBRE DE GROUPES) x (TEMPS MAXIMAL D'UTILISATION DE LA RESSOURCE PAR GROUPE).

Si ce chien de garde détecte un dépassement de temps, il signale qu'un autre type d'erreur s'est produit dans l'anneau à passage de jeton.

6.2.7 Gestionnaire d'écriture avec accusé de réception

Un gestionnaire d'écriture avec accusé de réception (HWC) est représenté par la figure 6-17 ci-dessous. Il est similaire à un gestionnaire d'écriture associé à un indicateur, disposé de manière à fournir des fonctions d'acquiescement au niveau de la trame lorsqu'il est utilisé avec un écouteur transactionnel avec accusé de réception (HTL).

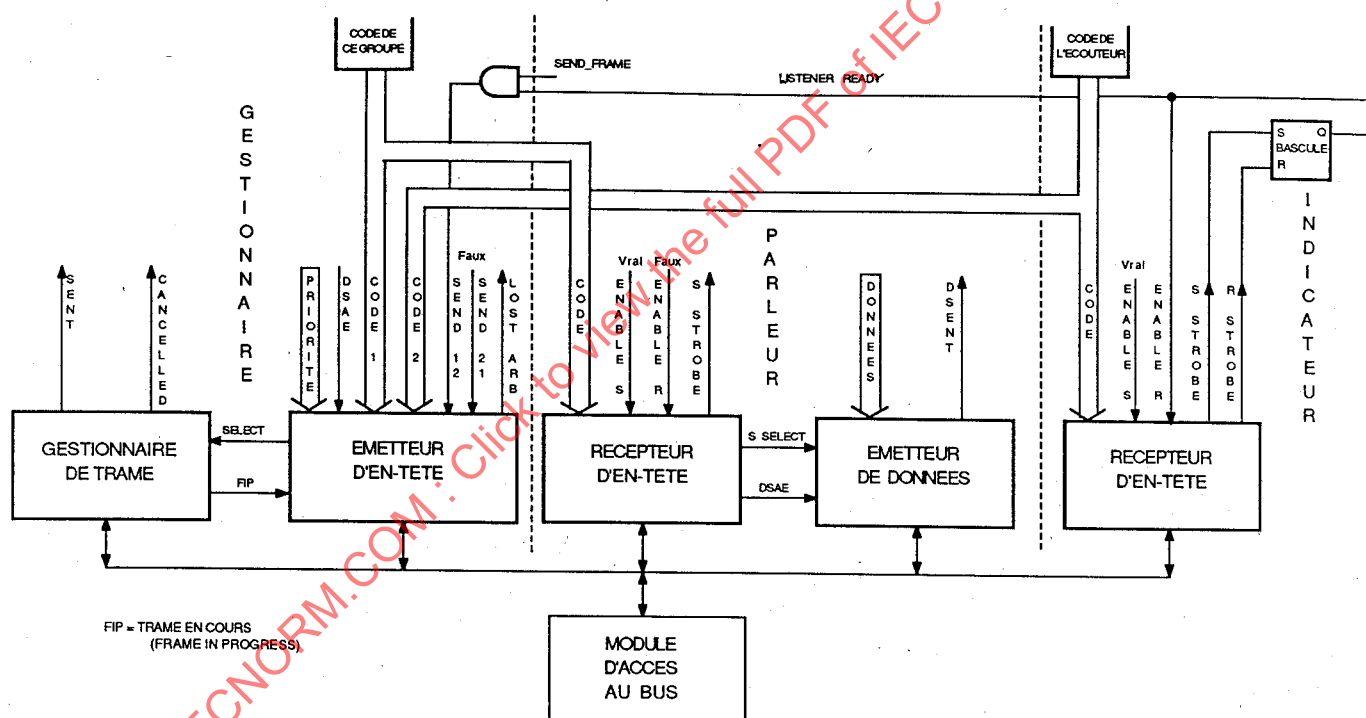


Figure 6-17 - Gestionnaire d'écriture avec accusé de réception

6.2.7.1 But

Les écouteurs transactionnels peuvent différer beaucoup par la vitesse selon laquelle ils traitent ou "utilisent" les données. Si un écouteur utilise normalement des données avant qu'une autre trame de données lui soit adressée, il n'invalidé alors une trame que rarement ou même jamais et le bus série est utilisé efficacement. Si, cependant, un écouteur prend, normalement trop de temps pour traiter des données alors que d'autres trames lui parviennent avant qu'il ne soit prêt à les recevoir, alors il invalidera ces trames. Les invalidations ne sont pas une utilisation productive du bus série. Dans cette dernière situation, les groupes à "accusé de réception" peuvent être utilisés pour empêcher que des trames soient invalidées.

RECOMMANDATION 6.5:

En plus de la vérification d'erreur permise par la logique traitant RING ERROR, il convient que les couches supérieures gèrent un chien de garde qui est déclenché lorsqu'une trame à jeton est reçue, et dont l'intervalle temporel est de:

(NOMBRE DE GROUPES) x (TEMPS MAXIMAL D'UTILISATION DE LA RESSOURCE PAR GROUPE).

Si ce chien de garde détecte un dépassement de temps, il signale qu'un autre type d'erreur s'est produit dans l'anneau à passage de jeton.

6.2.7 Gestionnaire d'écriture avec accusé de réception

Un gestionnaire d'écriture avec accusé de réception (HWC) est représenté par la figure 6-17 ci-dessous. Il est similaire à un gestionnaire d'écriture associé à un indicateur, disposé de manière à fournir des fonctions d'acquiescement au niveau de la trame lorsqu'il est utilisé avec un écouteur transactionnel avec accusé de réception (HTL).

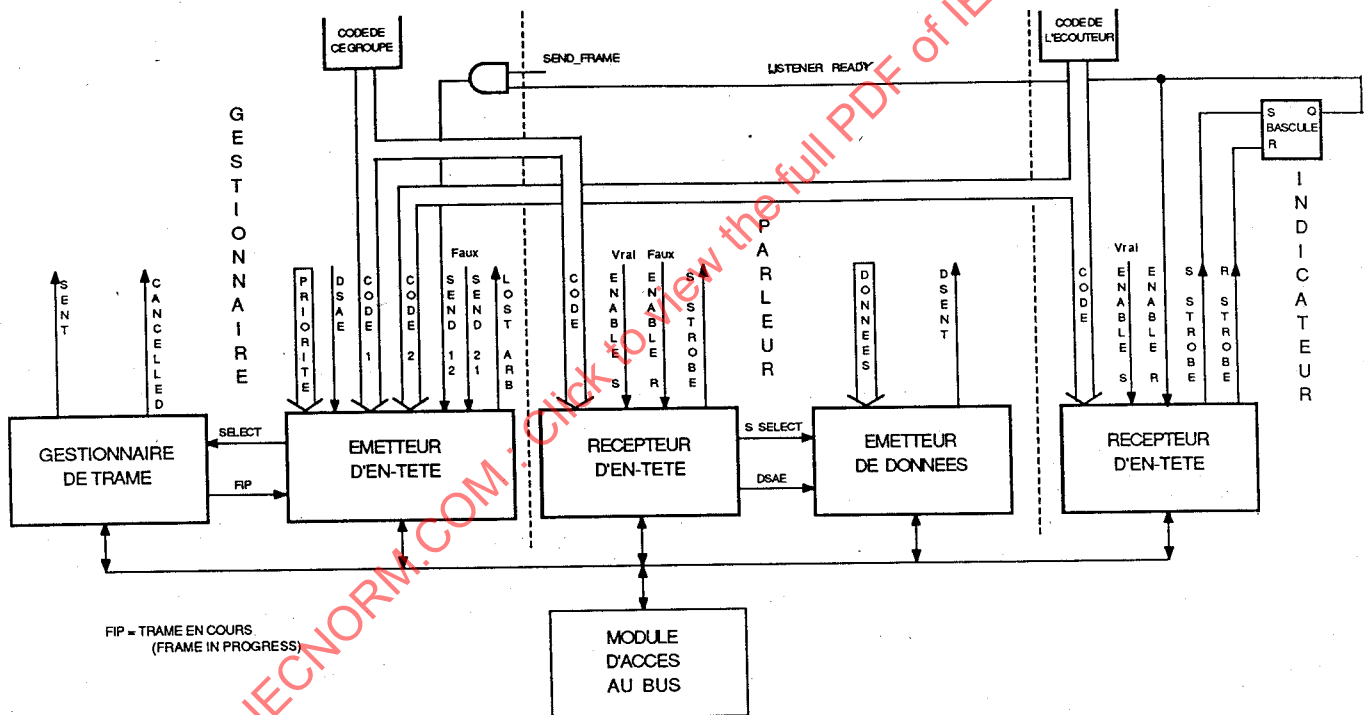


Figure 6-17 - Gestionnaire d'écriture avec accusé de réception

6.2.7.1 But

Les écouteurs transactionnels peuvent différer beaucoup par la vitesse selon laquelle ils traitent ou "utilisent" les données. Si un écouteur utilise normalement des données avant qu'une autre trame de données lui soit adressée, il n'invalidé alors une trame que rarement ou même jamais et le bus série est utilisé efficacement. Si, cependant, un écouteur prend, normalement trop de temps pour traiter des données alors que d'autres trames lui parviennent avant qu'il ne soit prêt à les recevoir, alors il invalidera ces trames. Les invalidations ne sont pas une utilisation productive du bus série. Dans cette dernière situation, les groupes à "accusé de réception" peuvent être utilisés pour empêcher que des trames soient invalidées.

Un gestionnaire d'écriture avec accusé de réception est utilisé pour émettre des trames de données multiples à un écouteur transactionnel à accusé de réception, à un taux commandé par la rapidité avec laquelle les couches supérieures de l'écouteur traitent ou "utilisent" les données. Cela est réalisé en faisant renvoyer par l'écouteur transactionnel à accusé de réception une trame de réponse au gestionnaire d'écriture avec accusé de réception, lui permettant ainsi d'émettre une autre trame de données.

6.2.7.2 Description

Le gestionnaire et le parleur d'un gestionnaire d'écriture avec accusé de réception diffèrent de ceux utilisés dans le gestionnaire d'écriture décrit précédemment par une seule caractéristique: l'entrée SEND 12 de l'EMETTEUR D'EN-TETE n'est pas commandée directement par les couches supérieures du gestionnaire du bus série, mais est conditionnée par la sortie de l'indicateur associé. Le RECEPTEUR D'EN-TETE de l'indicateur répond à l'adresse de l'écouteur transactionnel avec accusé de réception auquel la donnée est envoyée. Lorsqu'une trame de transfert de données est émise, la sortie R STROBE du RECEPTEUR D'EN-TETE met à 0 la bascule et positionne le signal LISTENER READY à l'état faux. Cela empêche que le signal SEND_FRAME en provenance des couches supérieures ne rende SEND 12 à l'état vrai à l'EMETTEUR D'EN-TETE et ne déclenche une autre trame de données, jusqu'à ce que l'écouteur transactionnel avec accusé de réception renvoie une trame de réponse avec son adresse dans le champ S. La trame de réponse met alors à un l'indicateur et positionne LISTENER READY à l'état vrai afin qu'une autre trame de données puisse être émise.

6.2.7.3 Fonctionnement

Lorsque les couches supérieures du gestionnaire du bus série associées à un gestionnaire d'écriture avec accusé de réception décident d'émettre une séquence de trames à un écouteur transactionnel avec accusé de réception, elles configurent le gestionnaire d'écriture avec accusé de réception comme suit:

- 1) Elles positionnent les ports de priorité et d'arbitrage d'émission de données de l'EMETTEUR D'EN-TETE selon les trames à émettre.
- 2) Elles positionnent le port de code 1 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE du parleur à l'adresse affectée à ce groupe.
- 3) Elles positionnent le port de code 2 de l'EMETTEUR D'EN-TETE et le port de code du RECEPTEUR D'EN-TETE de l'indicateur à l'adresse affectée à l'écouteur transactionnel avec accusé de réception auquel la donnée doit être transmise.
- 4) Elles mettent l'entrée ENABLE S du RECEPTEUR D'EN-TETE de l'écouteur à l'état vrai.
- 5) Elles mettent à un la bascule de l'indicateur, rendant LISTENER READY à l'état vrai, et par conséquent, l'entrée ENABLE R du RECEPTEUR D'EN-TETE à l'état vrai.
- 6) Elles mettent l'entrée ENABLE S du RECEPTEUR D'EN-TETE de l'indicateur à l'état vrai.

A Handshaking Writing Controller is used to send multiple frames of data to a Handshaking Transaction Listener, at a rate controlled by how fast the Listener's higher layers process or "dispose of" data. This is accomplished by having the HTL send a response frame back to the HWC, thereby permitting it to send another data frame.

6.2.7.2 Description

The Controller and Talker in a Handshaking Writing Controller differ from those in a Writing Controller as described earlier, in one regard only: the SEND 12 input to the HEADER SENDER is not controlled directly by higher layers of serial bus management, but is conditioned by the output of the associated Flag. The Flag's HEADER RECEIVER responds to the address of the HTL to which data is sent. When a data transfer frame is sent, the HEADER RECEIVER's R STROBE output resets the latch and makes the LISTENER READY signal False. This prevents the SEND_FRAME signal from higher layers from making SEND 12 True to the HEADER SENDER and initiating another data frame, until the HTL sends a response frame with its address in the S field. The response frame then sets the Flag and makes LISTENER READY True so that another data frame can be sent.

6.2.7.3 Operation

When higher layers of serial bus management with a Handshaking Writing Controller decide to send a sequence of frames to a Handshaking Transaction Listener, they configure the HWC as follows:

- 1) They set the HEADER SENDER's Priority and Data Sender Arbitration Ports as appropriate for the frames to be sent.
- 2) They set the HEADER SENDER's Code 1 Port and the Code Port of the Talker's HEADER RECEIVER to the address assigned to this group.
- 3) They set the HEADER SENDER's Code 2 Port and the Code Port of the Flag's HEADER RECEIVER to the address assigned to the HTL to which the data is to be sent.
- 4) They make the ENABLE S input of the Talker's HEADER RECEIVER True.
- 5) They set the Flag's latch, making LISTENER READY True and thus the ENABLE R input of the Flag's HEADER RECEIVER True.
- 6) They make the ENABLE S input of the Flag's HEADER RECEIVER True.