
Transmissions séries

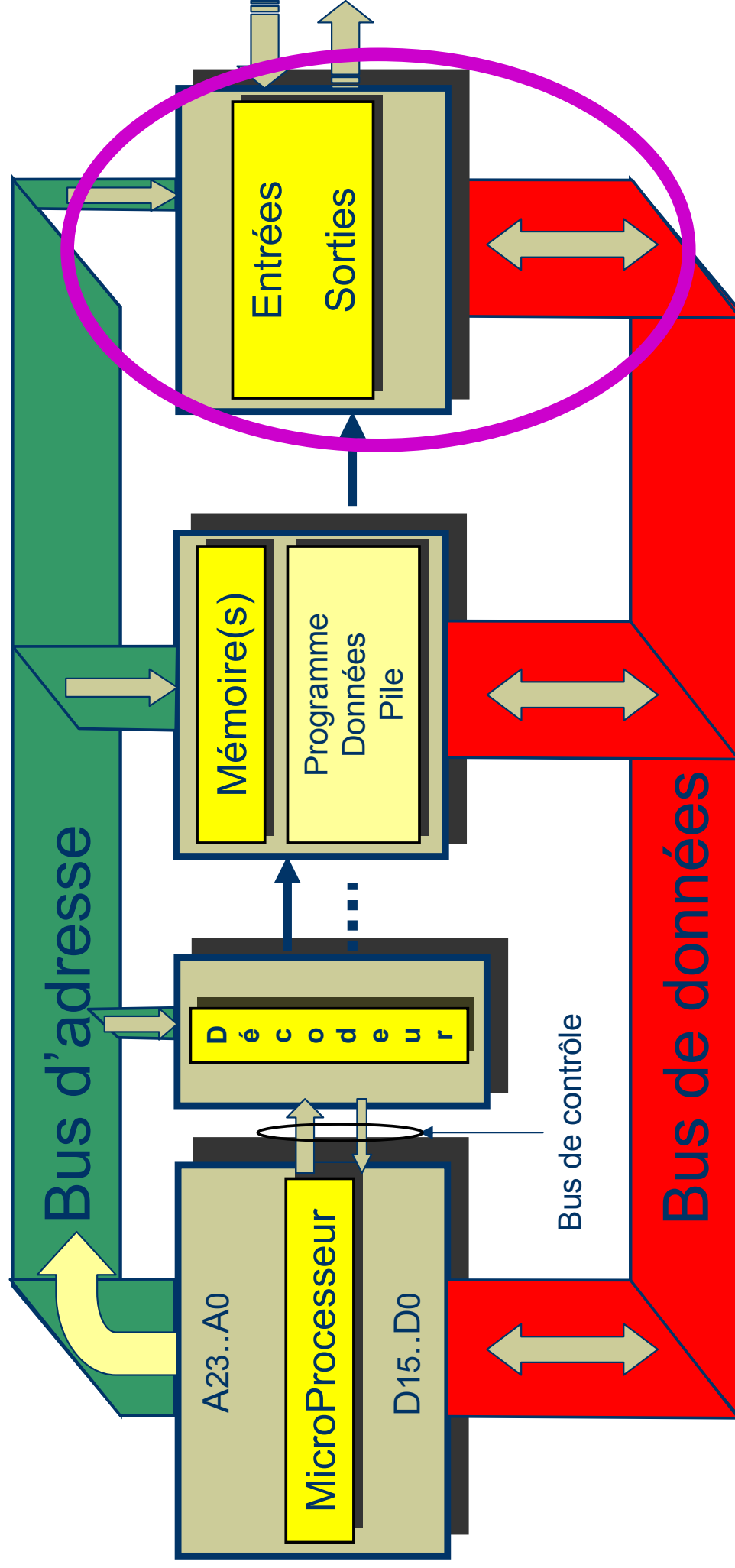
Base des transmissions séries

René Beuchat

LAP/EPFL

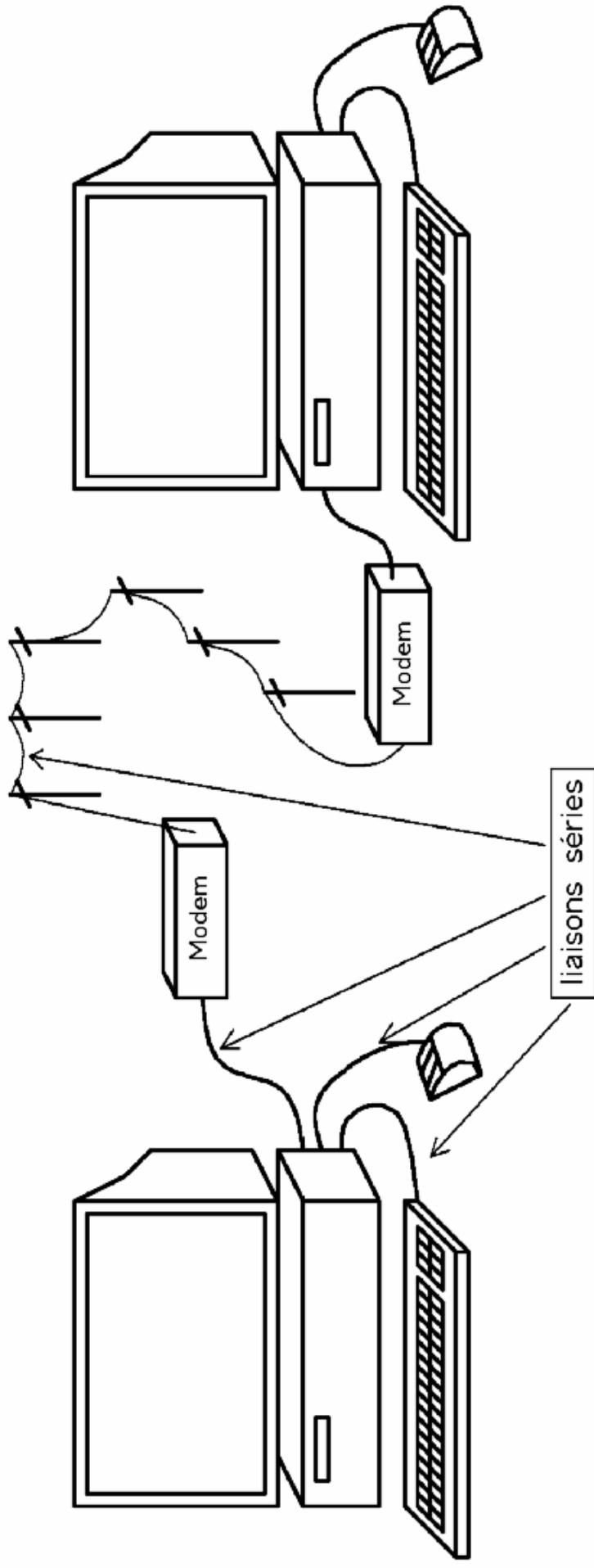
LSN/EIG/HESO

Architecture d'un système informatique



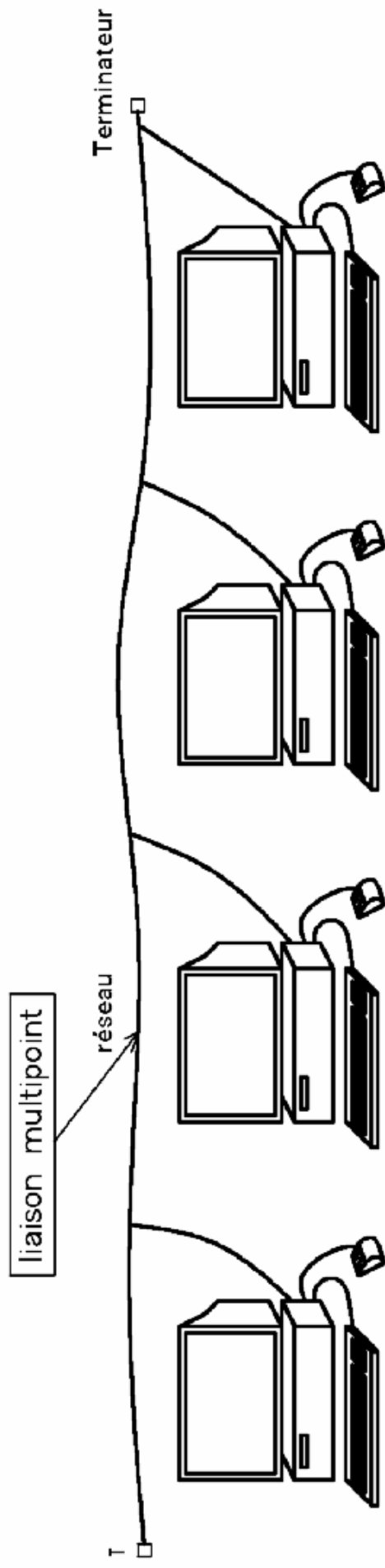
Liaisons séries sur un ordinateur

- Clavier, souris (PS2, USB)
- Modem (RS232, V24)
- Ligne téléphonique (analogique, numérique)



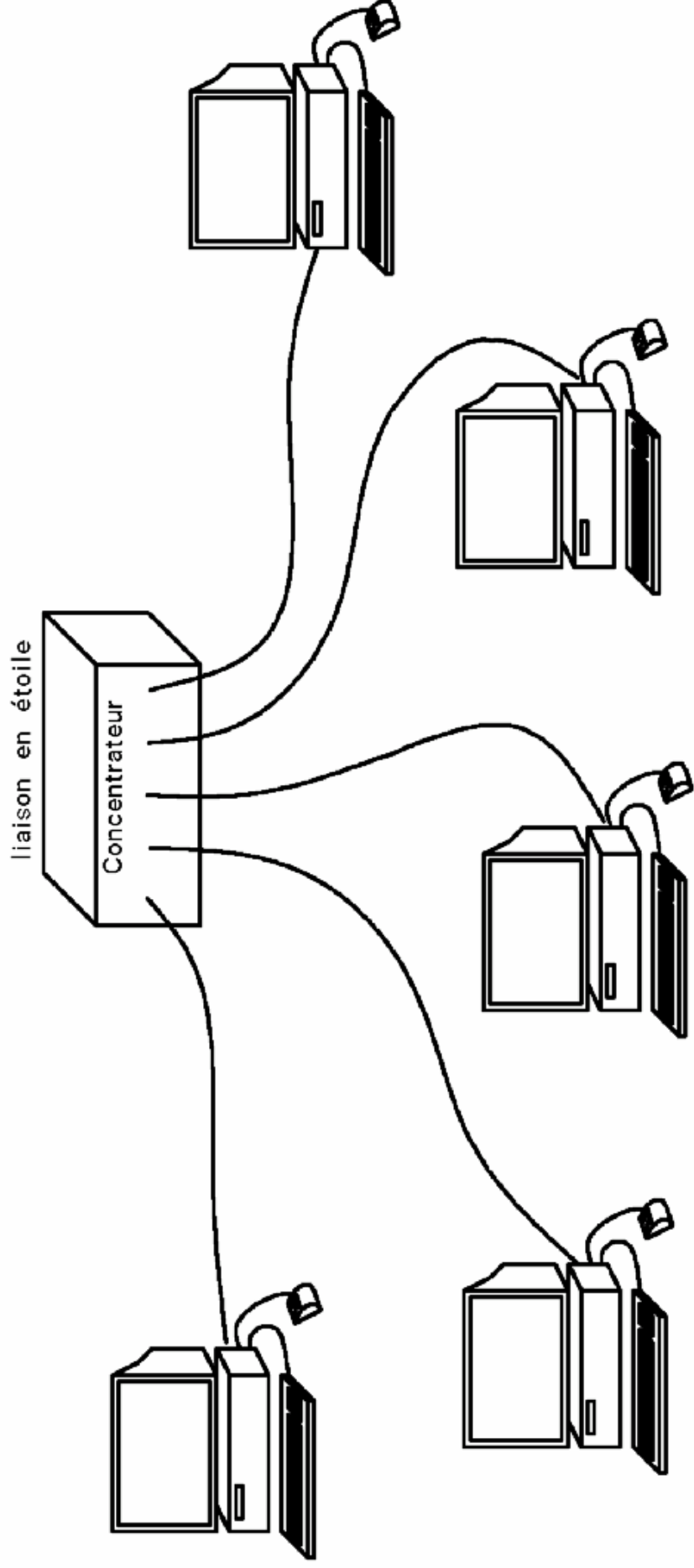
Liaisons séries sur un ordinateur

- Liaisons multipoint (Ethernet 10 base2)
- Partage du même câble par plusieurs utilisateurs



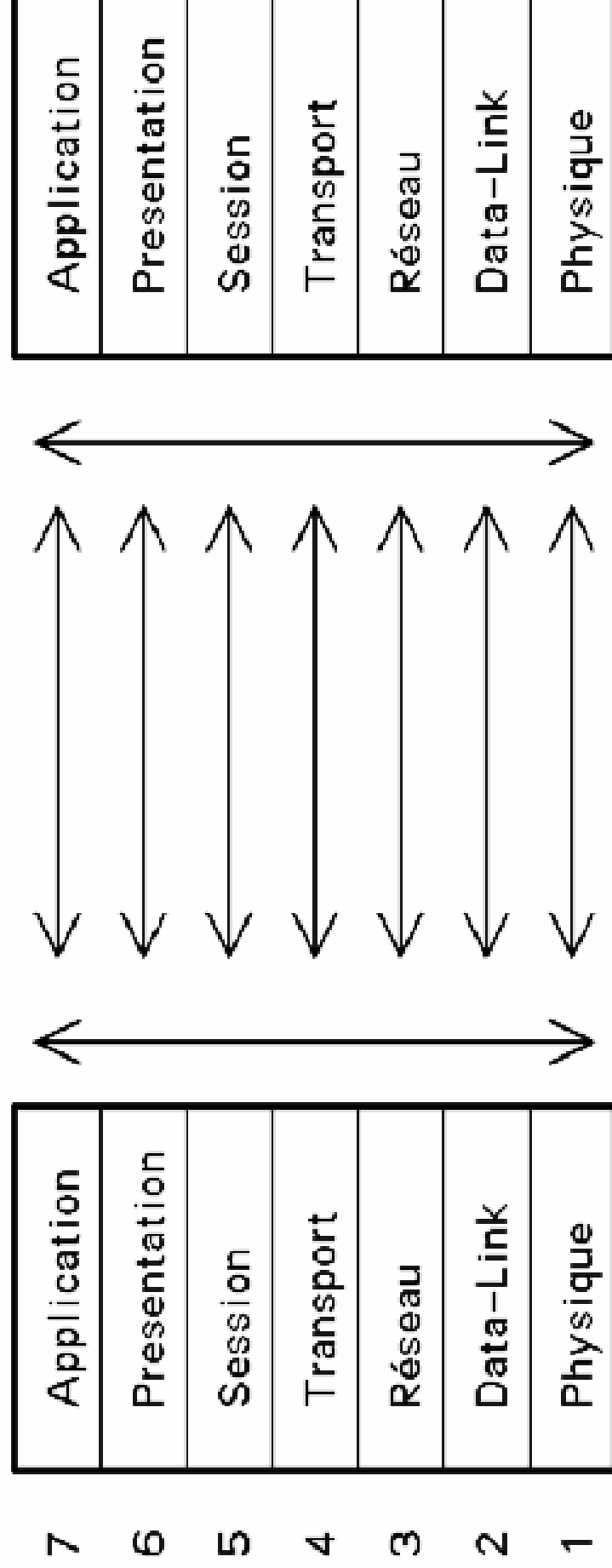
Liaisons séries sur un ordinateur

- Liaisons point à point (Ethernet 10/100 Twisted)
- Liaison en étoile



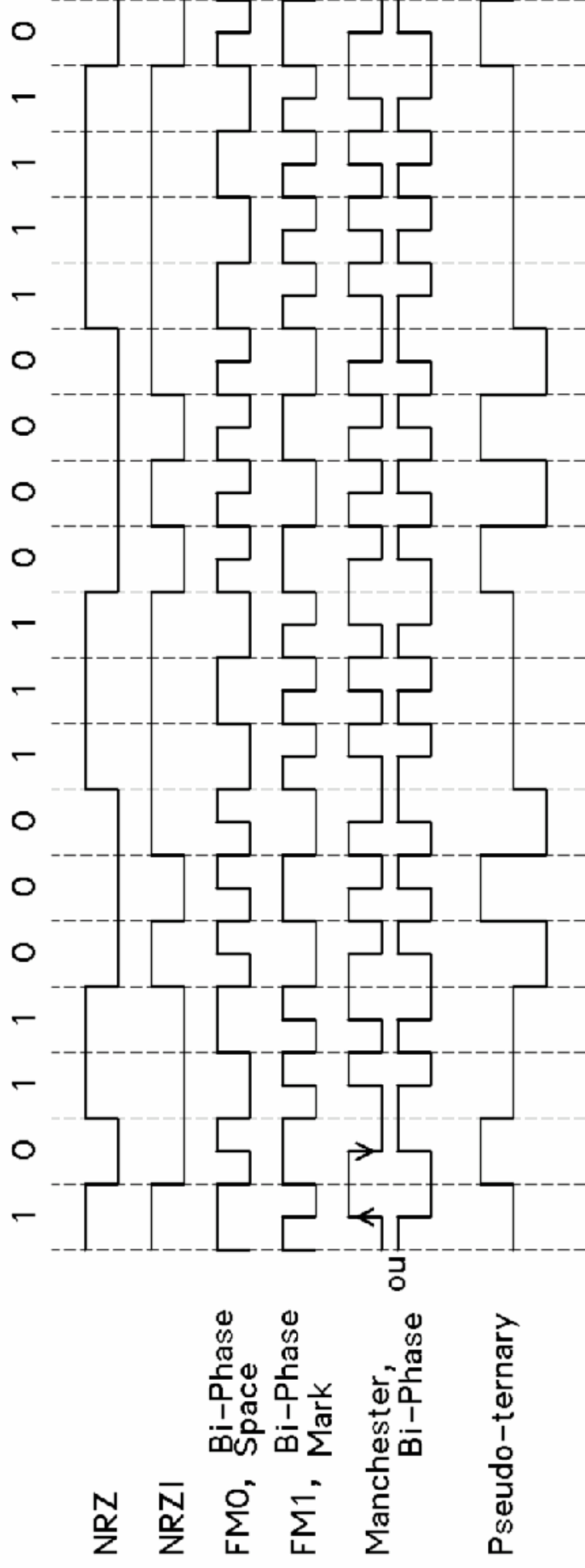
7 couches OSI de l'ISO

- Open System Interconnection
- International Standard Organization



Codage de bits

Exemples de codages

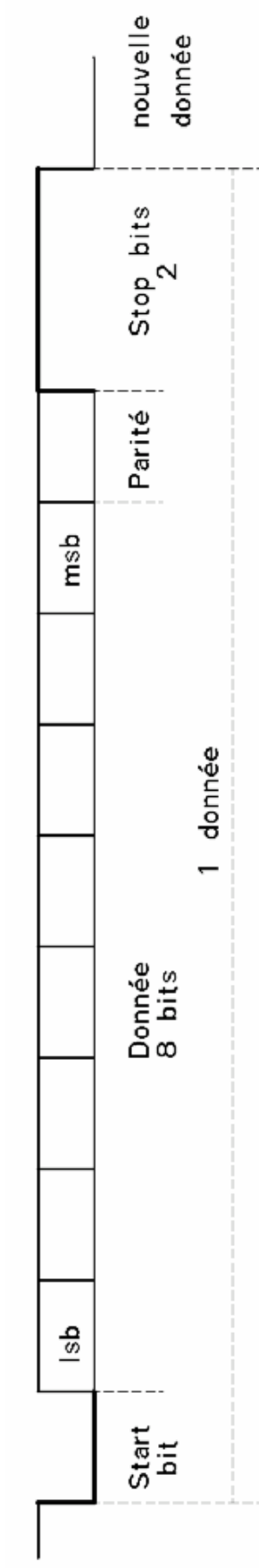


Synchronisations

Problèmes de synchronisations :

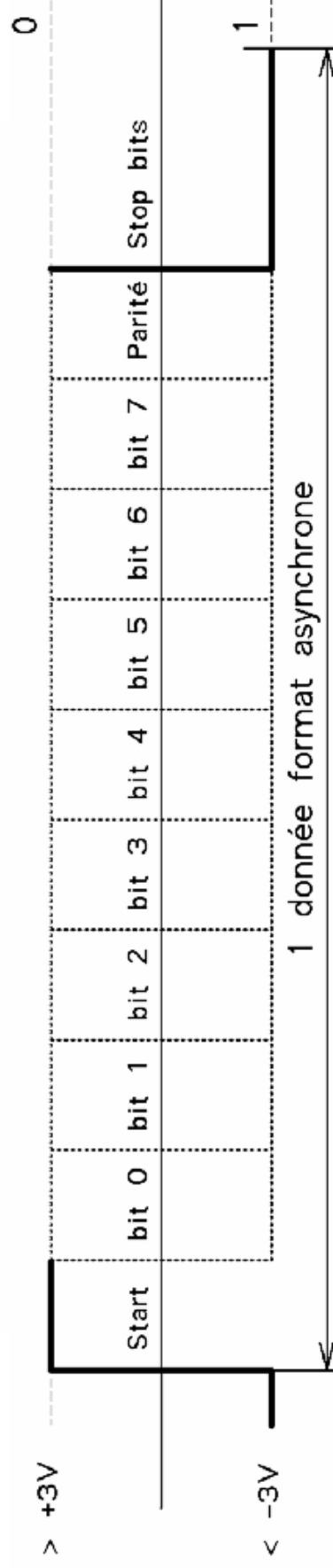
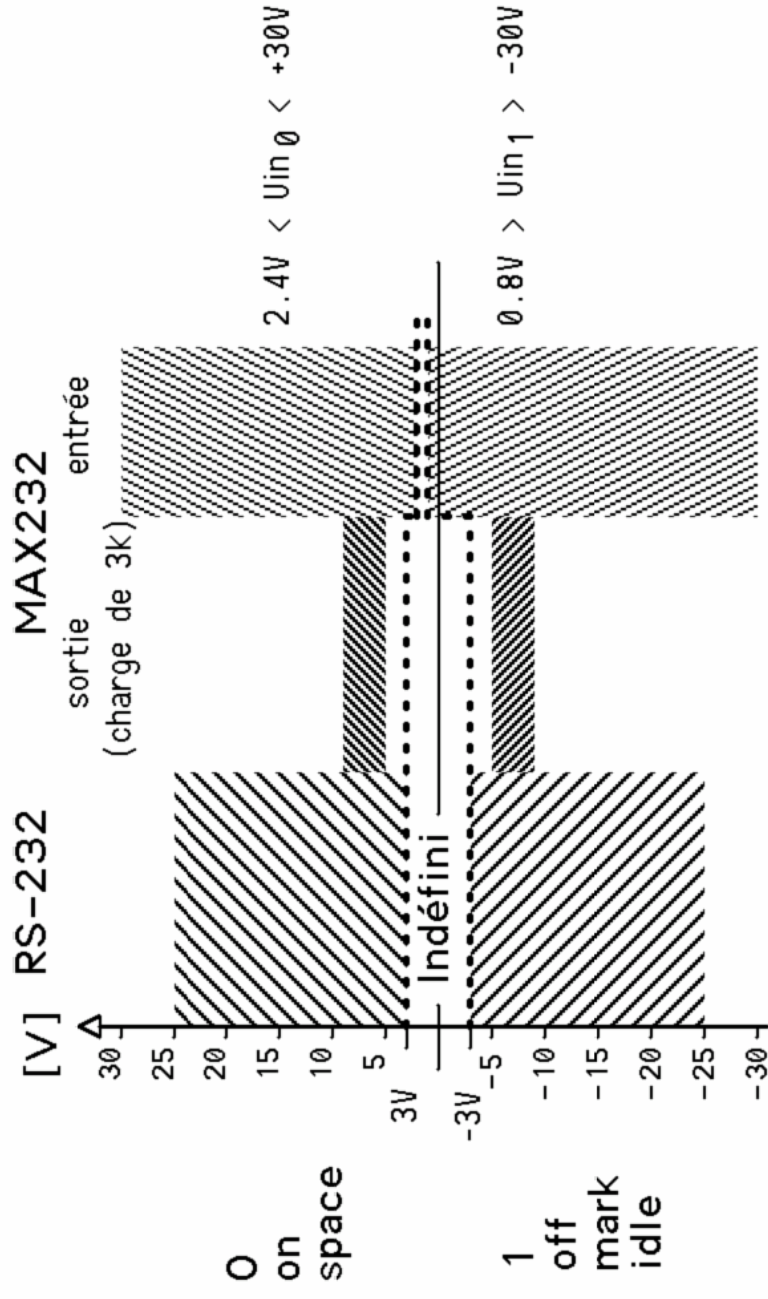
- Niveau bit
- Niveau mot (assemblage de bits)
- Niveau message (assemblage de mots)
- Vitesse de transmission

Transmission asynchrone



- Synchronisation par mot :
 - start bit
 - donnée (5..8 bits)
 - parité (facultatif, paire/impaire)
 - stop bit (1, 1.5, 2 au minimum)
- Vitesse de transmission PAR CHOIX EXTERNE

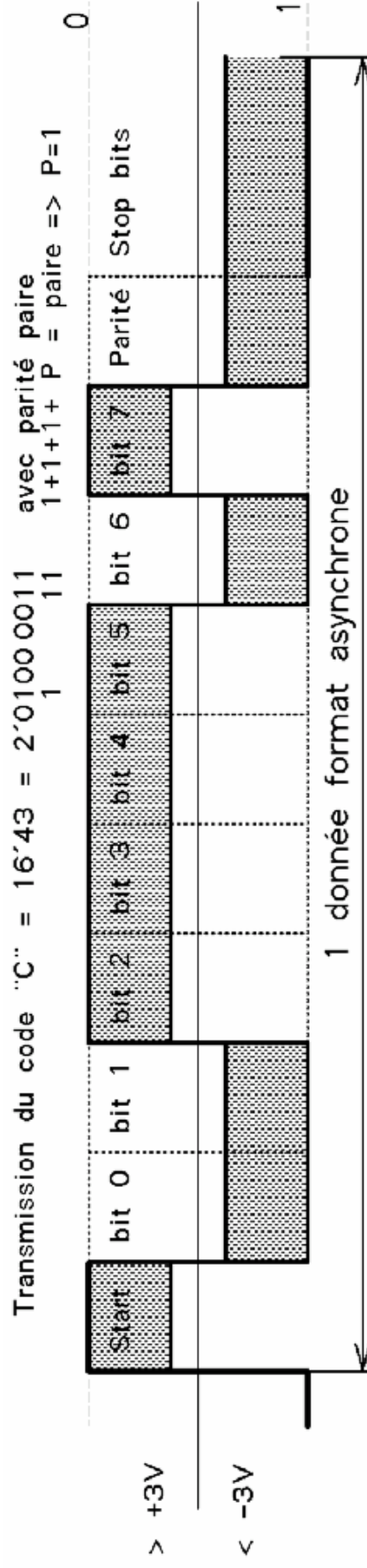
Transmission asynchrone, RS-232



Exemple, RS-232

Transmission code "C", parité paire :

- ASCII C \rightarrow 16'43 \rightarrow 2' 0100 0011
- Lsb (bit 0) en premier
- '0' \rightarrow > +3V
- '1' \rightarrow < -3V



Exemple, RS-232

- 3 signaux sont indispensables pour la communication:
 - **Gnd**: Terre de transmission (pin 7)
 - **TxD**: transmission de données
 - **RxD**: réception de données
- Viennent ensuite 2 signaux de gestion du contrôle de flux de transmission entre l'émetteur et le récepteur:
 - **RTS**: *Requets To Send*, l'émetteur désire émettre
 - **CTS**: *Clear To Send*, le récepteur autorise l'émetteur à émettre
- Un signal est utilisé pour indiquer que la communication est établie:
 - **DCD**: *Data Carrier Detect*, la porteuse est valide
- 2 signaux indiquent que les équipements sont prêts pour communiquer:
 - **DTR**: *Data Terminal Ready*, le terminal est prêt
 - **DSR**: *Data Set Ready*, le modem est prêt
- Un dernier signal utilisé avec certains modems:
 - **RI**: *Ring Indicator*, sonnerie

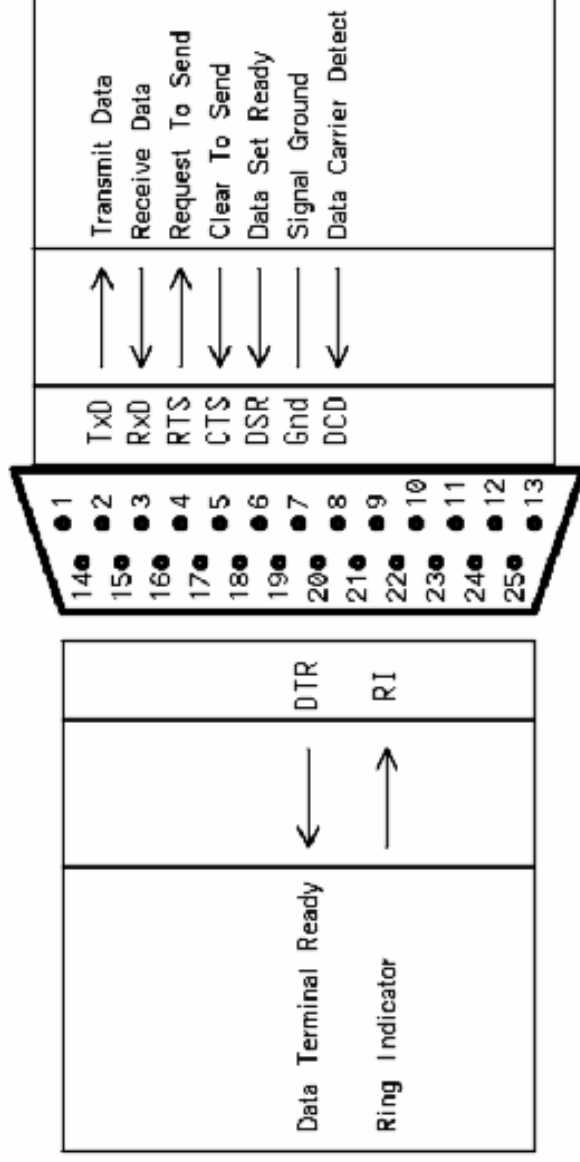
Connecteur RS-232, 25 pôles

Description	Abréviation	direction DCE DTE	EIA RS-232	CCITT V24						direction DTE DCE	Abréviation	Description
Transmission données (cs)	Sec. Transmitted Data	←	SBA	118	EDS	STxD	14●	1●	ED	→	Transmit Data	Terre de protection
Horloge émission interne	Tr. signal element timing	→	DB	114	HE	??	15●	2●	RD	←	Receive Data	Transmission de données
Réception données (cs)	Sec. Received Data	→	SBB	119	RDS	SRxD	16●	3●	DPE	→	Request To Send	Demande pour émettre
Horloge de réception	Rec. signal element timing	→	DD	115	HR	??	17●	4●	PAE	←	Clear To Send	Prêt à émettre
Requête de transm. (cs)	Sec. Request To Send	←	SCA	120	DPES	SRTS	18●	5●	PDP	←	Data Set Ready	Poste de données prêt
Terminal de données prêt	Data Terminal Ready	←	CE-CD	108	CPD-TDP	DTR	19●	6●	Gnd	→	Signal Ground	Terre de signalisation
Qualité du signal reçu	Signal Quality Detector	→	CG	110	QSR-QUA	SQD	20●	7●	DCD	←	Data Carrier Detect	Détection de porteuse
Indicateur d'appel	Ring Indicator	→	-	125	IA	RI	21●	8●	-			
Sélection de débit binaire	Data Signal Rate Selector	↔	CH-CI	111/112	SDB	DSRS	22●	9●	-			
Horloge émission externe	Tr. signal element timing	←	DA	113	HEE	TSET	23●	10●	SFE	→	Select Transmit Frequency	Sélection de fréq. d'émiss.
							24●	11●	DPS	←	Secondary Carrier Detect	Détection de porteuse (cs)
							25●	12●	BSS	←	Secondary Clear To Send	Prêt à émettre (cs)
								13●	PAES	←		

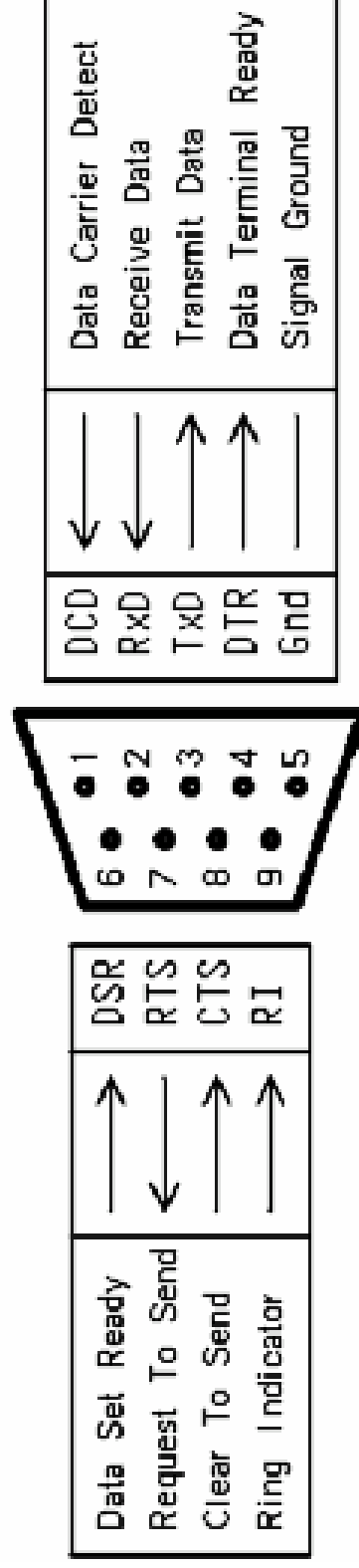
cs : canal secondaire

Connecteur RS-232, 9 pôles

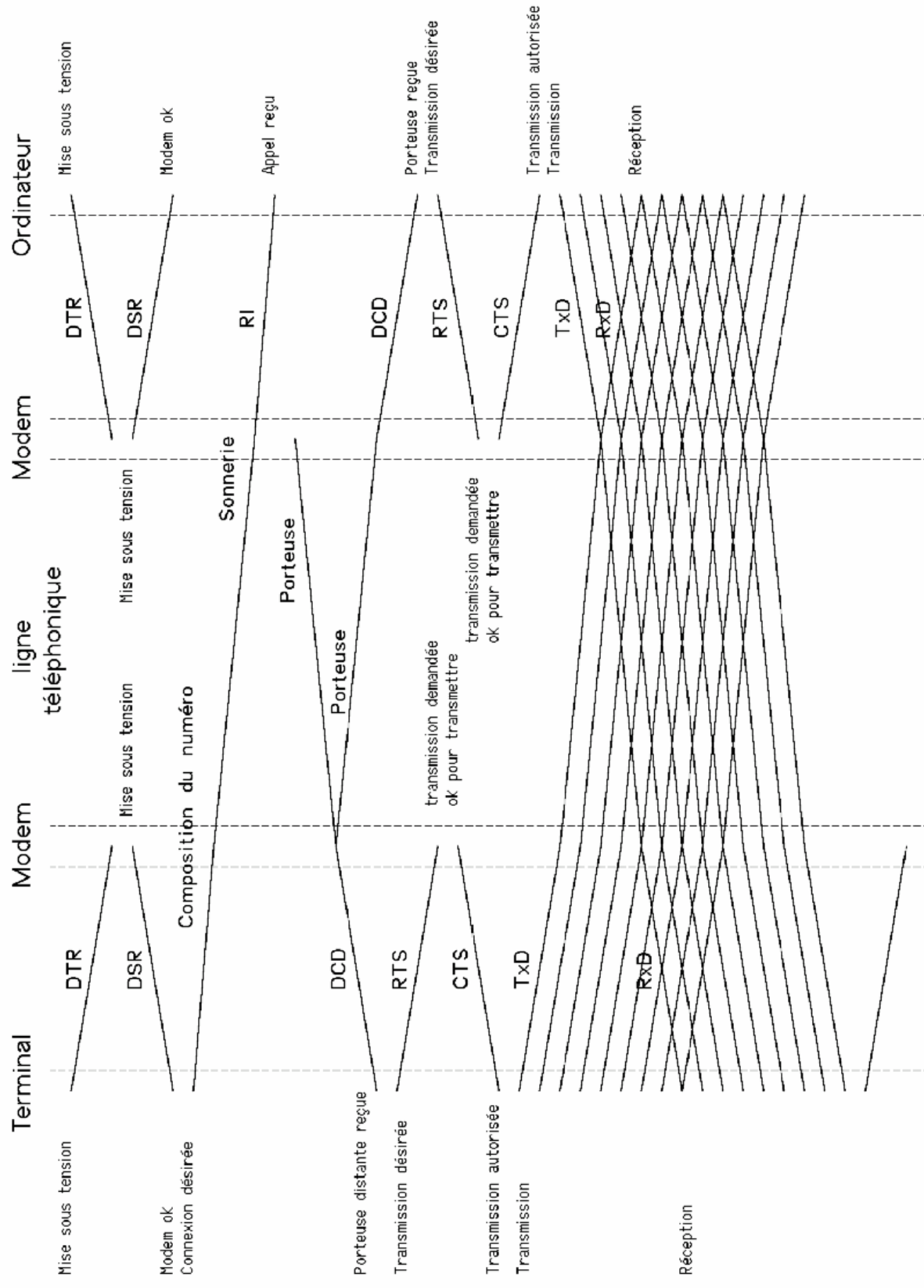
DB-25 mâle



DB-9 mâle

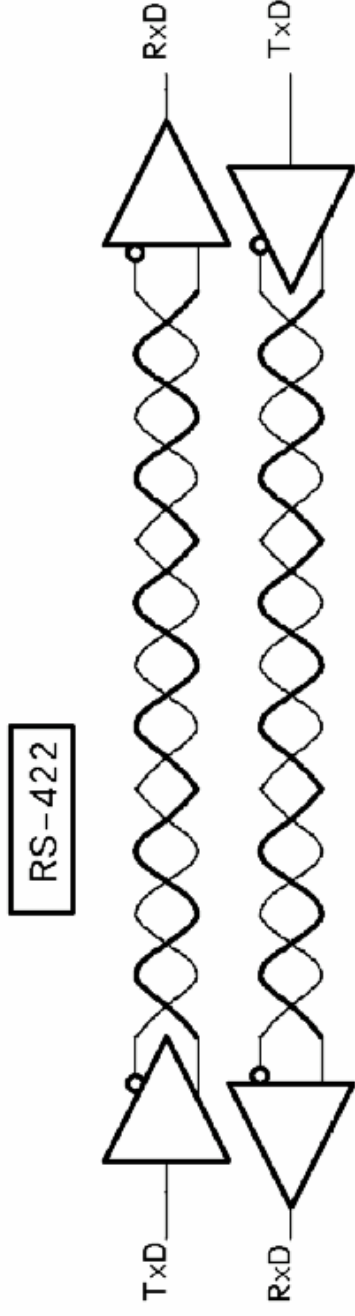


Dialogue RS-232

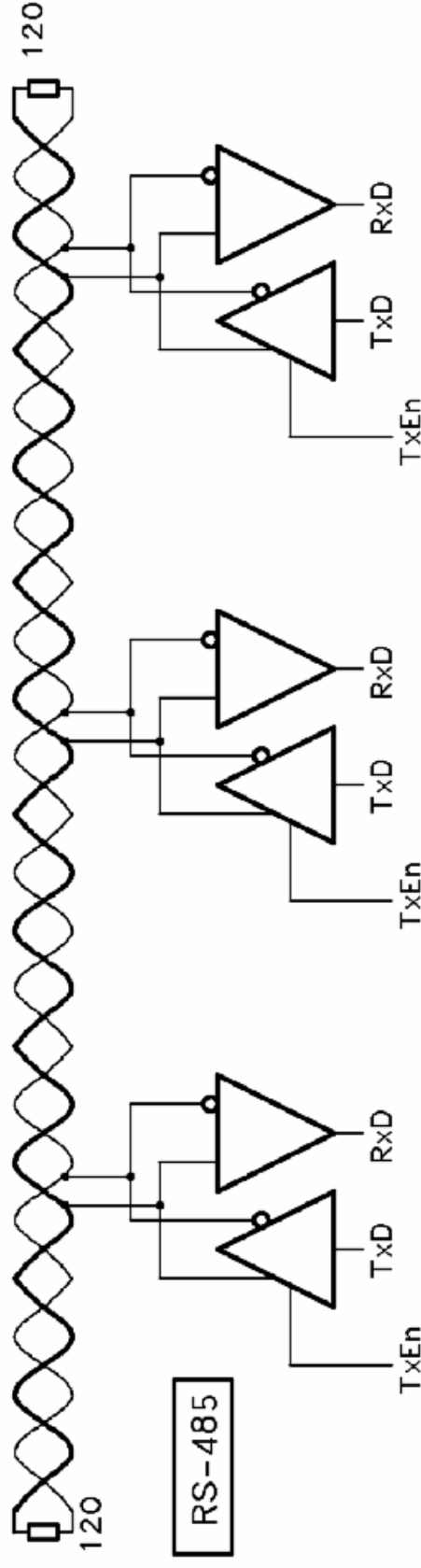


RS422- RS485, connexion physique

Transmission différentielle, point à point



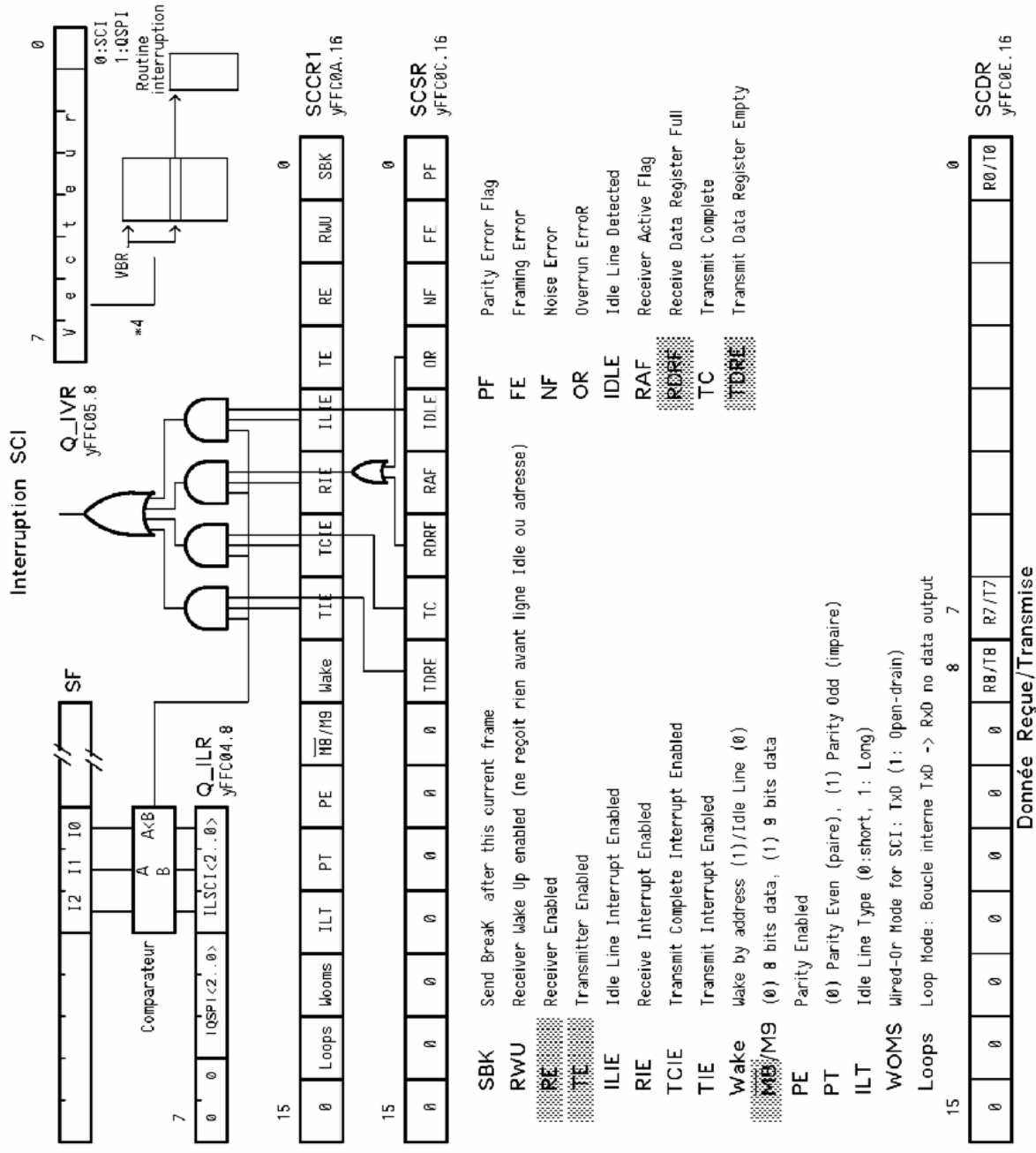
Transmission différentielle, multipoints



Exemple asynchrone

- ◆ Interface série asynchrone du 68331
- ◆ Module SCI, Serial Communication Interface
- ◆ Interface programmable intégrée sur 68331
- ◆ Programmation par registres de contrôle et de statuts
- ◆ Transmission de données par registre de donnée

Modèle interface série 68331



Programmation SCI (1)

Title SCI331_1

```
; Programme pour carte MAN331 (Manip 68331)
; LAMI-EPFL
; R.Beuchat 5.4.98, 29.4.99

; Utilisation interface SCI en émission par scrutation

; Adresses diverses
; *****

CSIO      EQU $FFFF8000      ; Adresse I0
MUBUS     EQU CSIO
Led        EQU CSIO+1<<6      ; Affichage de 6 Led (bits 5..0)
SwitchHexa EQU CSIO+1<<6
Stack     EQU $280000        ; Pile

; Adresses SCI
; -----

Q_SCBR      EQU $FFFFFFC08      ; Acces 16 bits 15..0
Q_SCCR1     EQU $FFFFFFC0A      ; Acces 8 bits 7..0
Q_SCCR1L    EQU Q_SCCR1+1
Q_SCDR      EQU $FFFFFFC0E
Q_SCDRL     EQU Q_SCDR+1
Q_SCSR      EQU $FFFFFFC0C
bTE         EQU 3
bTDRE       EQU 8
CLKSYS      EQU 16777216
SCI_BaudEQU EQU 9600
```

Programmation SCI (2)

Programme principal

- Initialisation SCI en transmission
- Boucle à l'infini :
 - Emission du code lu sur Mubus (interrupteurs)

```
XDEF      _Startup

_Startup:  Move.l    #Stack,SP
           Move.w    #...,Q_SCCR
           Move.w    #...,Q_SCCR1

           bSet.b    #bTE,Q_SCCR1low
Loop:      Move.b    Mubus,D0
           Bsr       SndSCI
           Bra       Loop

           ; Initialise pointeur de pile
           ; Initialise Baud Rate (9600)
           ; Rien de spécial, pas d'interruption

           ; Transmission autorisée
           ; Lecture Mubus -> donnée à transmettre

           ; Et on boucle
```

Programmation SCI (3)

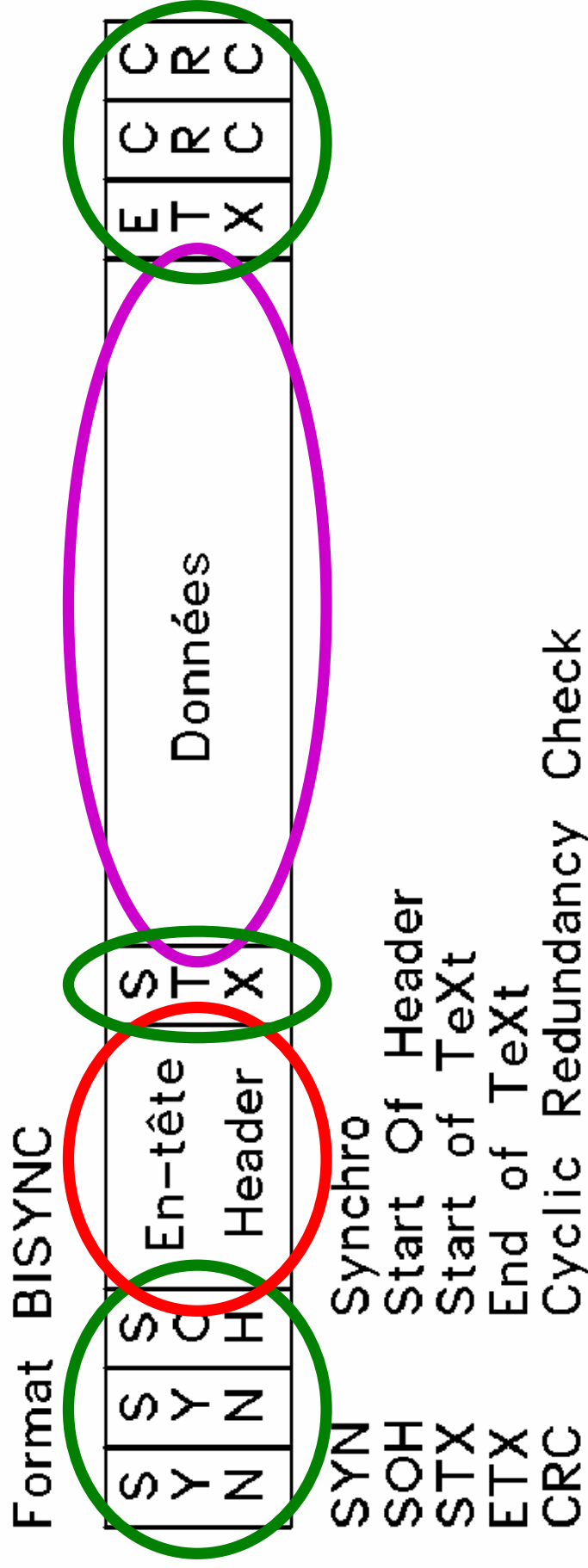
Sous-programme émission d'un octet

```
;-----\
; SndSCI > Emission d'une donnée sur ligne série
;=====/
; In:  D0.b    donnée à transmettre 8 bits
; Out:  -
; Mod:  -

SndSCI:
LSnd:
    ...      LSnd
    BEQ      D0,Q_SCDR+1
    Move.b   D0,Q_SCDR+1
    RTS

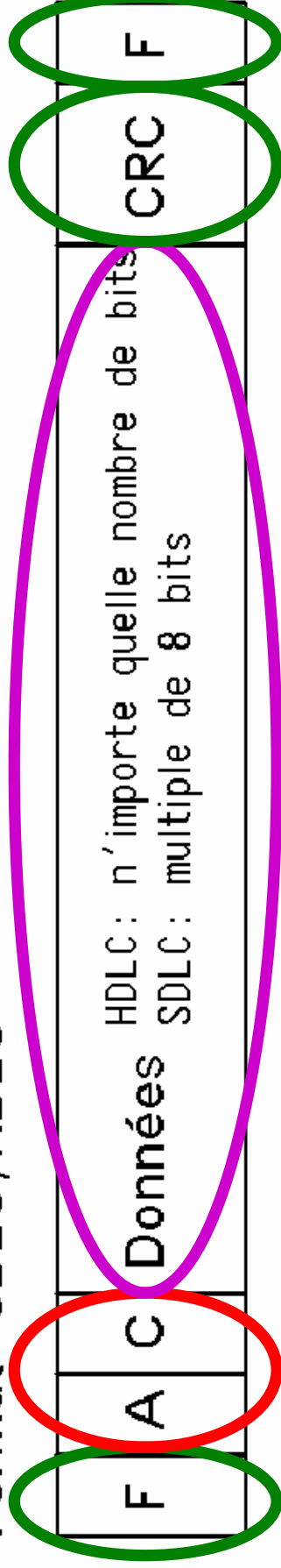
    ; Transmission possible ?
    ; non -> attend
    ; oui -> transmet 8 bits
    ; !! Q_SCDR définit sur 16 bits !!
```

Transmission par paquet (1), BISYNC



Transmission par paquet (2), HDLC

Format SDLC/HDLC



HDLC: n'importe quelle nombre de bits
SDLC: multiple de 8 bits

F Flag 01111110

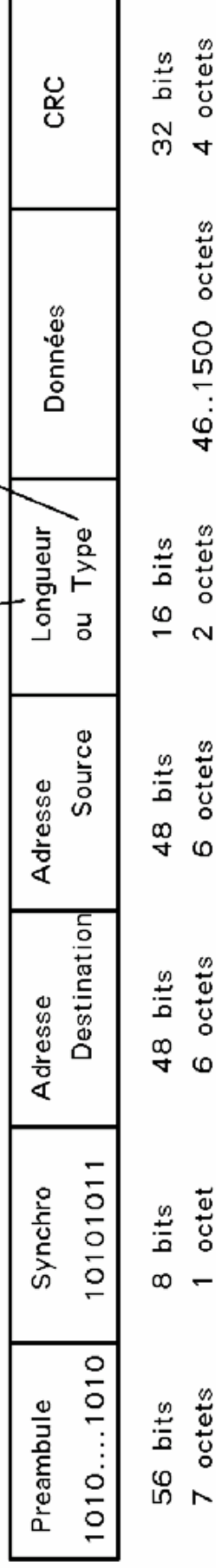
A Adresse destination (8 bits)

C Contrôle (8 bits)

Ethernet/IEEE 802.3

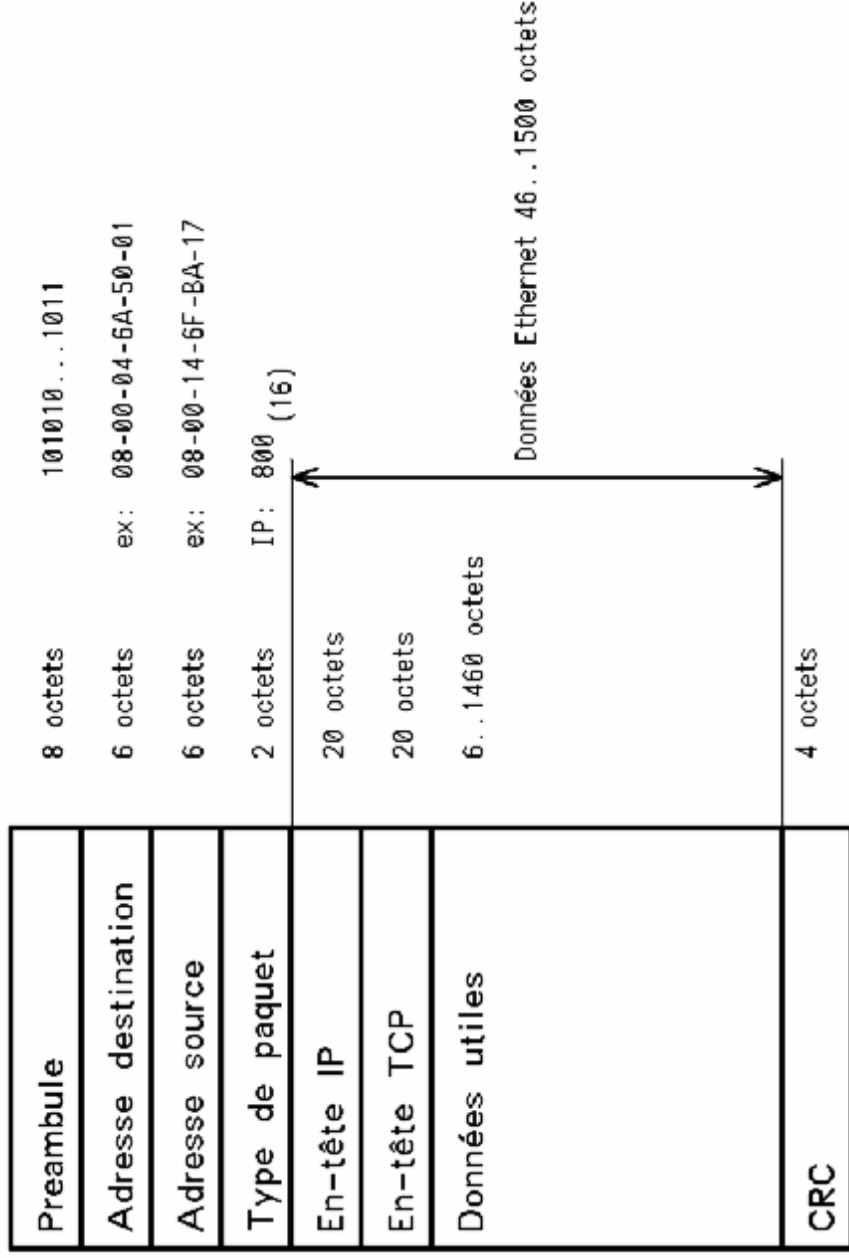
- Transmission par paquet
- Horloge codée avec données
- Codage Manchester

Paquet IEEE802.3/Ethernet



Ethernet/IEEE 802.3

- TCP/IP



Paquet avec TCP/IP

Interfaces séries

Bus I²C

Bus SPI

Bus 1-Wire

Transmission synchrone

Transmission de l'horloge et des données, ex :

- I²C, transmission par paquet
- SPI, transmission par mot (8 bits, QSPI 16 bits)

Interfaces séries

- ◆ Exemples de composants
 - Interface asynchrone Modèle interface série 68331
 - Interface SCI 68331
 - Circuit i-Wire, uLAN Mémoire Dallas/maxim uLAN
 - Mémoire série DS2224
 - Circuit I²C I2C, exemple PCF8574 port //
 - Port parallèle PCF8574 (vieux composants)

Interfaces séries

Bus I²C

SMBus

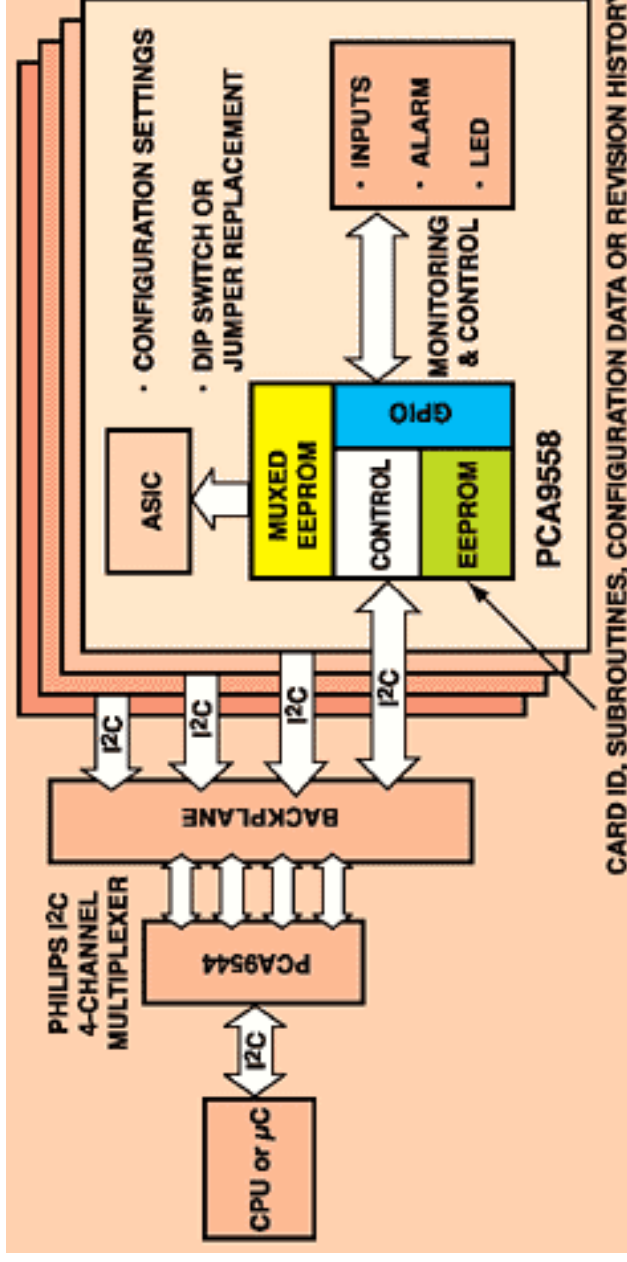
(System Management Bus)

Philips semiconductor

I²C

- ◆ Bus intercircuits
- ◆ Courte distance (~1 m)
- ◆ Multi-mâîtres
- ◆ Multi-esclaves
- ◆ Horloge fournie par le maître
- ◆ Bus collecteur ouvert
- ◆ Transmission à
 - 100 kbits/s (normal), 400 kbits/s (fast)

I²C, exemple d'utilisations



- ◆ Extension de ports I/O par simple bus
- ◆ série: port //, convertisseur A/D, D/A, mémoire série, ctrl tuning, etc...
- ◆ Ex. d'utilisation : chaîne hi-fi, TV, etc...

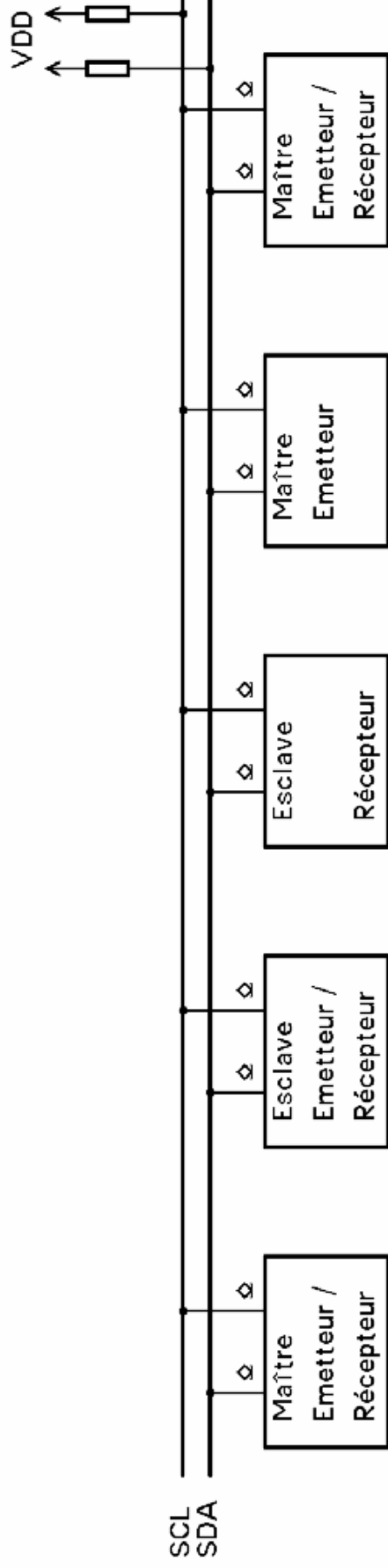
I²C, définitions

- ♦ Maître (*master*) :
 - Unité qui émet l'horloge, initie et termine le transfert
 - Fournit l'adresse de l'esclave sélectionné et le sens de transfert du cycle
- ♦ Esclave (*slave*) :
 - Unité sélectionnée pour le transfert par un maître
 - Il reçoit l'adresse et le sens du transfert
 - Quittance, s'il est l'unité sélectionnée
- ♦ Système multi-maître (*multimaster system*) :
 - Système où plusieurs maîtres peuvent prendre le bus sans corrompre les données
- ♦ Arbitrage (*arbitration*)
 - Procédure pour prendre le bus si plusieurs maîtres désirent le faire en même temps

I²C, définitions

- ◆ **Emetteur (*emettor*):**
 - Unité qui transmet une donnée sur le bus
 - Ecriture : l'émetteur est le maître
 - Lecture : l'émetteur est l'esclave
- ◆ **Récepteur (*receptor*):**
 - Unité qui reçoit une donnée sur le bus
 - Ecriture : le récepteur est l'esclave
 - Lecture : le récepteur est le maître
- ◆ **Synchronisation (*synchronization*)**
 - Méthode pour synchroniser l'horloge entre plusieurs circuits

I²C, interconnexion d'unités



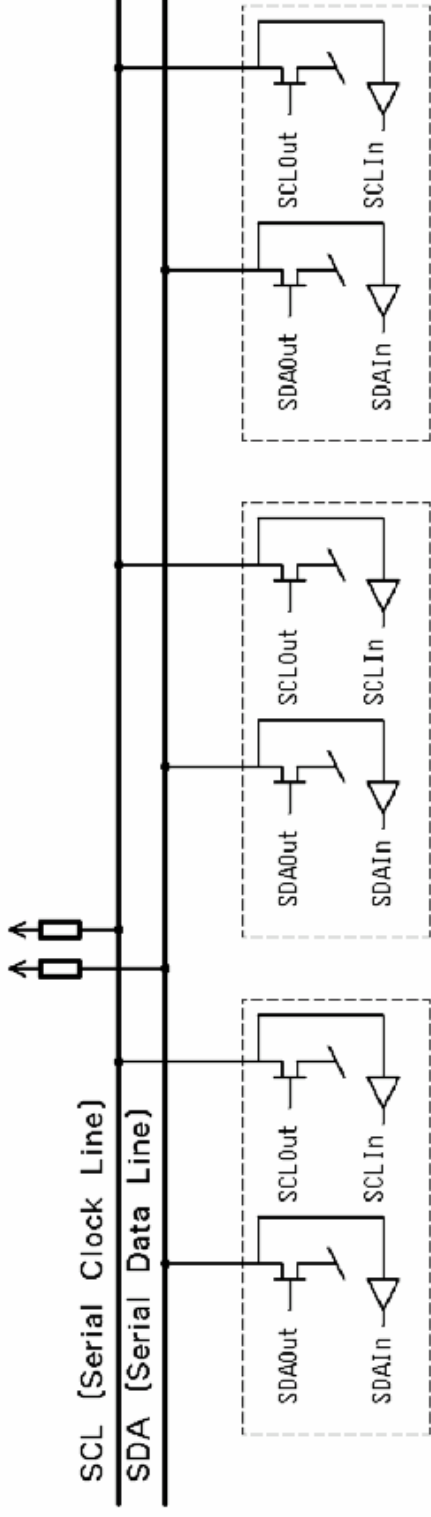
2 lignes pour tous :

- **SCL** : CLK, horloge
- **SDA** : Data, donnée

Transmission :

- multipoint
- synchrone
- par paquet

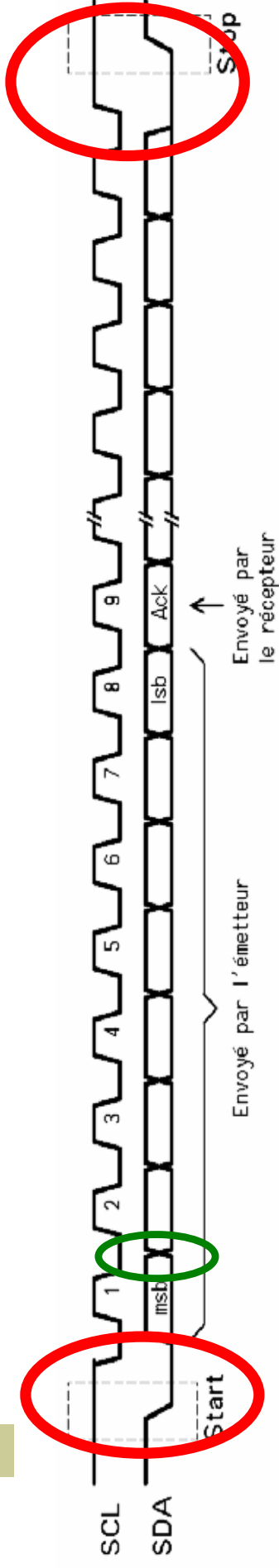
I²C, interface sur bus



- SCL : CLK, horloge,
- toujours activée par maître
- peut être maintenue à '0' par esclave
- SDA : Data, donnée

Lignes bidirectionnelles, **collecteur ouvert**

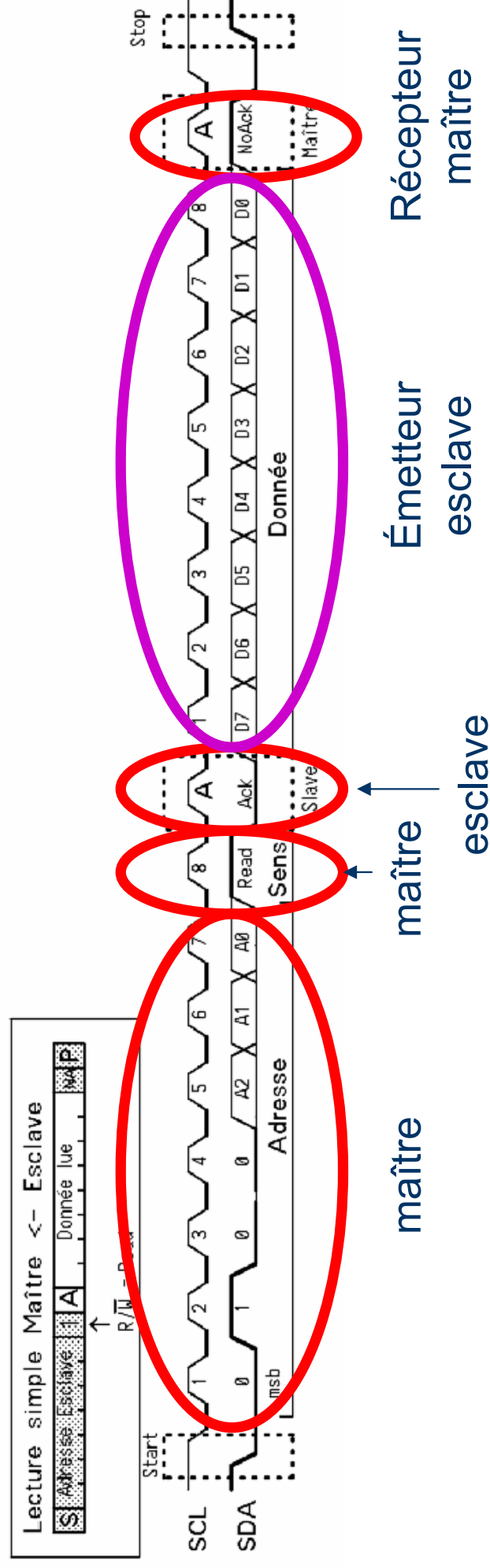
I²C, transmission d'un paquet START/STOP



- Repos lignes 'H'
- Start : SDA 'H' → '0', puis SCL 'H' → '0'
- Stop : SCL '0' → 'H', puis SDA '0' → 'H'
- Données changement avec SCL à '0'

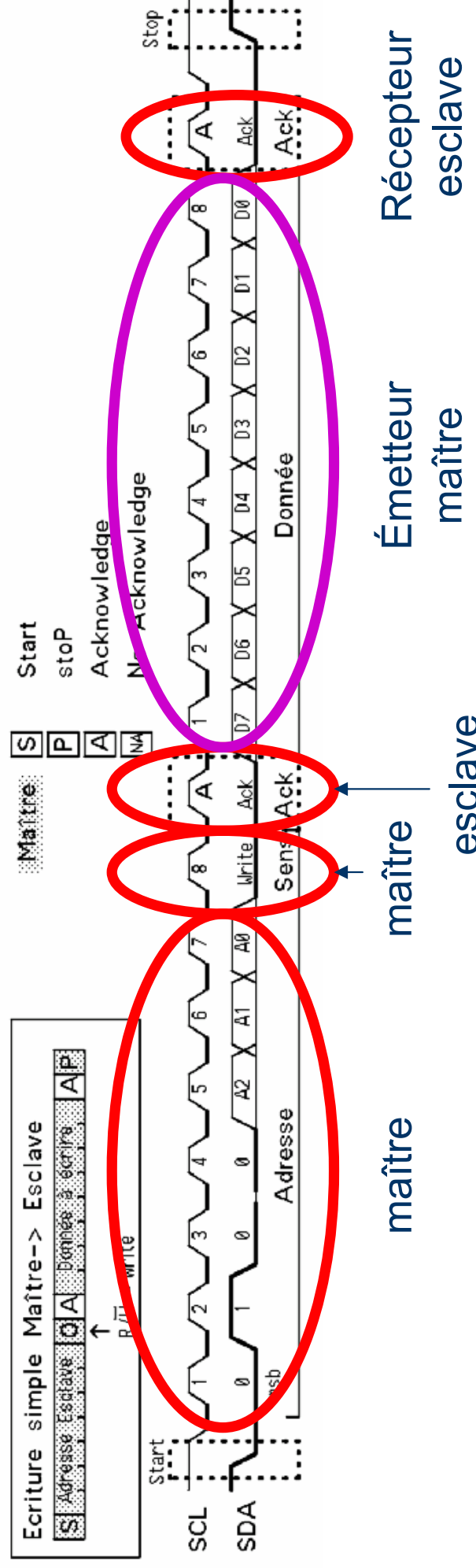
I²C lecture

- 1er octet :
- adresse destinataire 7 bits
- sens du transfert 1 bits
- Acknowledge (quittance)

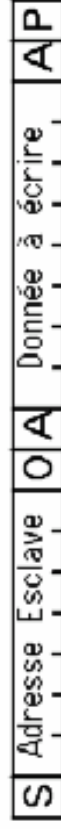


I²C écriture

- 1er octet :
- adresse destinataire 7 bits
- sens du transfert 1 bits
- Acknowledge (quittance)

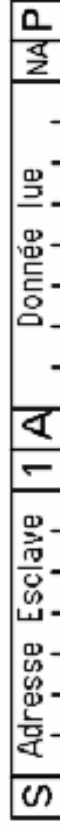


I²C autres accès



↑
R/ \overline{W} = Write

Ecriture simple Maître → Esclave



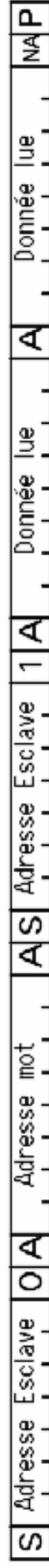
↑
R/ \overline{W} = Read

Lecture simple Maître ← Esclave



↑
R/ \overline{W} = Write

Ecriture multiple avec auto-incrémentation des adresses destination



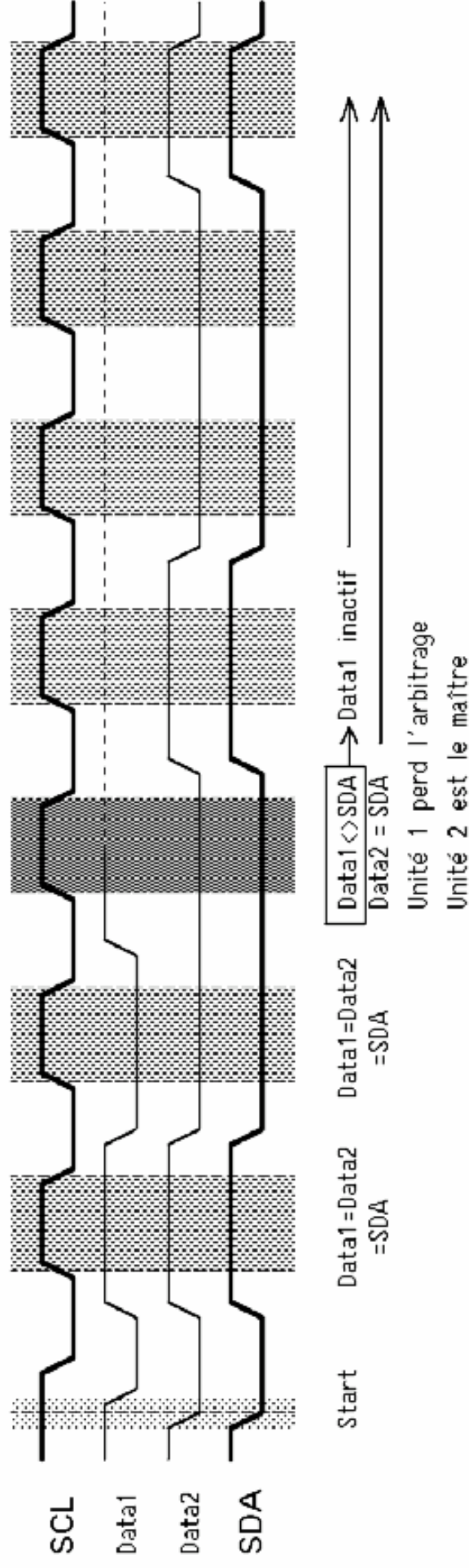
↑
R/ \overline{W} = Write

↑
R/ \overline{W} = Read

Lecture multiple avec auto-incrémentation des adresses destination

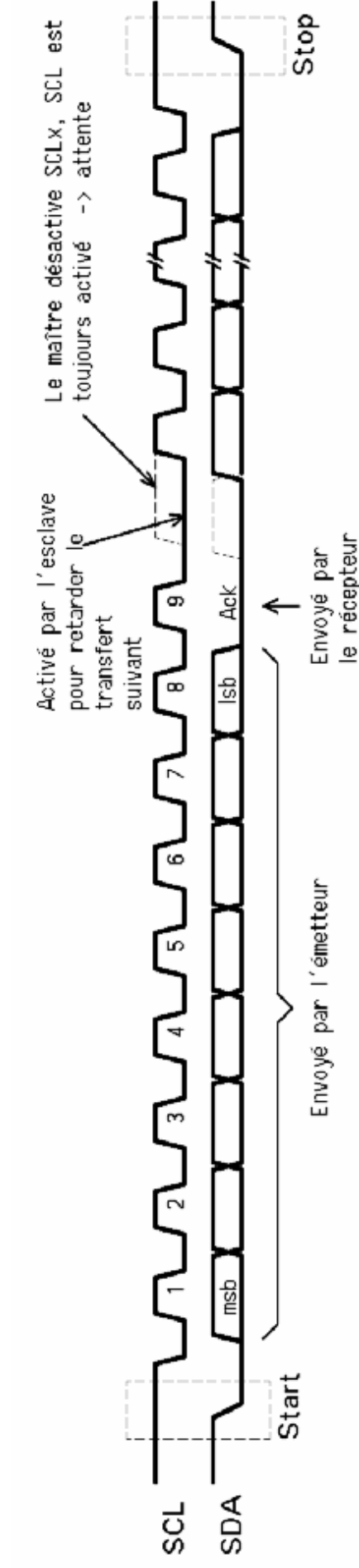
S Start
P stop
A Acknowledge
NA No Acknowledge
R/ \overline{W} Read si 1, Write si 0

I²C arbitrage multi-maîtres



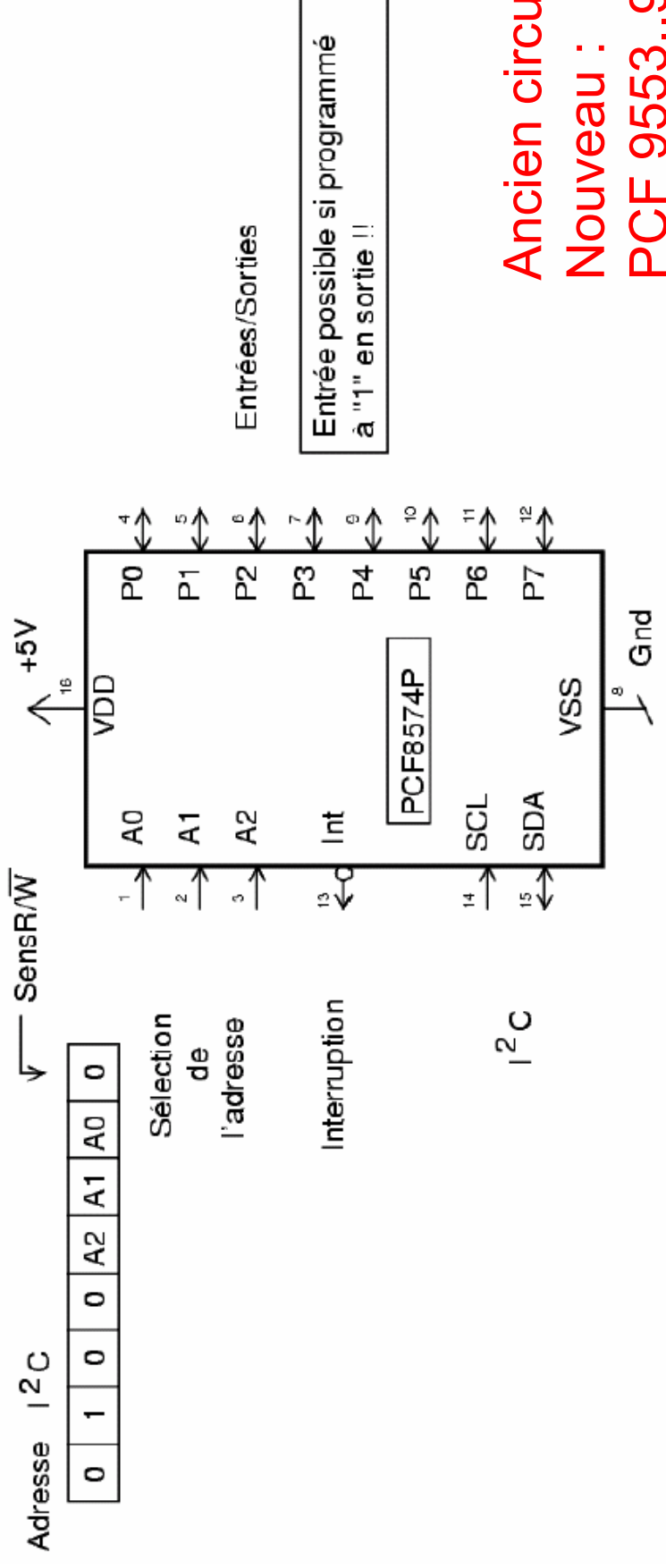
- Collecteur ouvert & relecture données transmises :
- SCL activé par tous les maîtres → '0'
- Mise de l'adresse destination SDA (msb → lsb)
- Dès que tous relâchent la ligne → 'H'
- Tant qu'une adresse destination a un bit à '0', maître correspondant continue la transmission

I²C Ralentissement de l'horloge



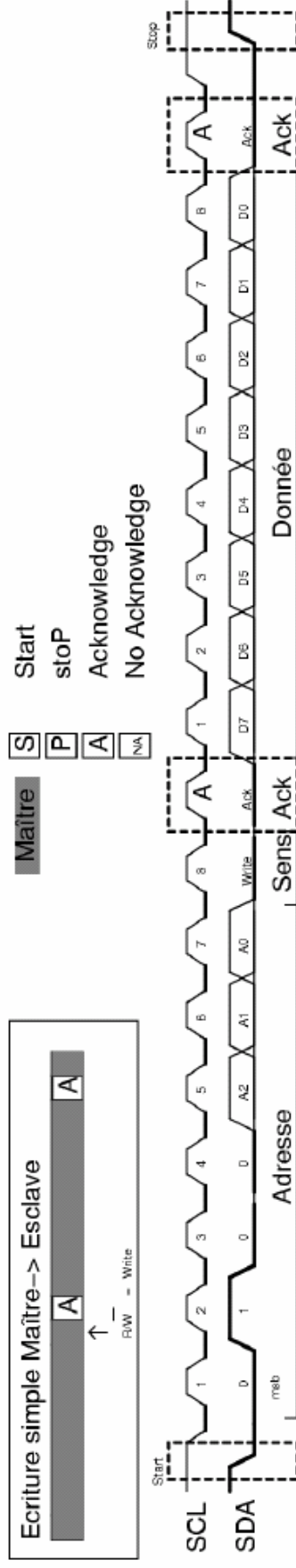
En maintenant l'horloge à '0', un esclave peut ralentir la transmission

I²C, exemple PCF8574 port //

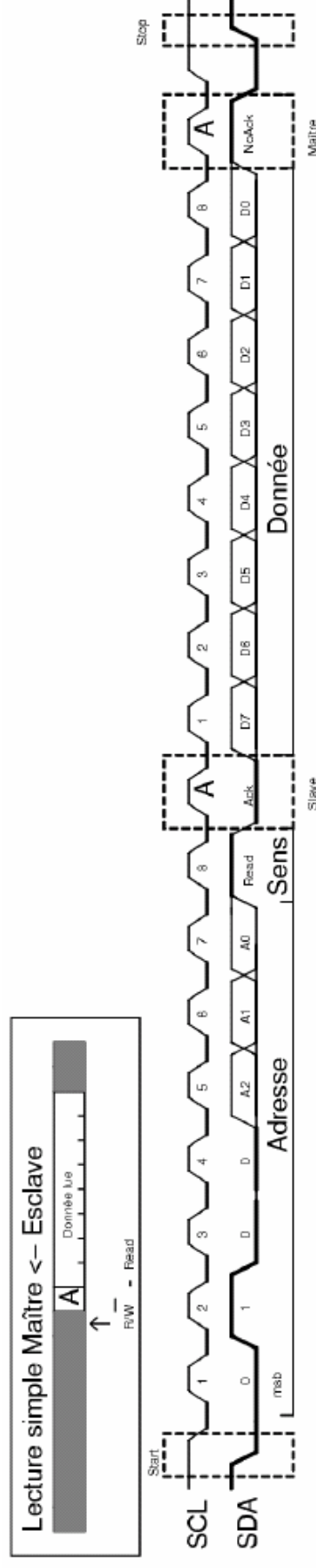


- ◆ Port parallèle 8 bits P7..P0
- ◆ 8 adresses sélectionnables par 3 entrées A2..A0
- ◆ 4 bits d'adresse fixes '0100'

I²C, exemple PCF8574 port //

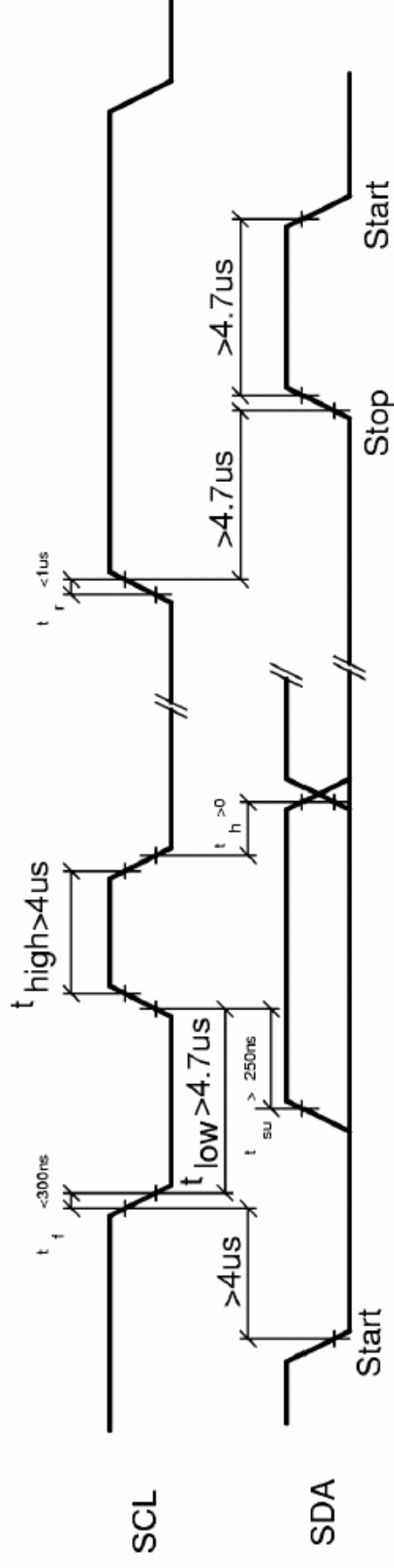


◆ Ecriture I²C

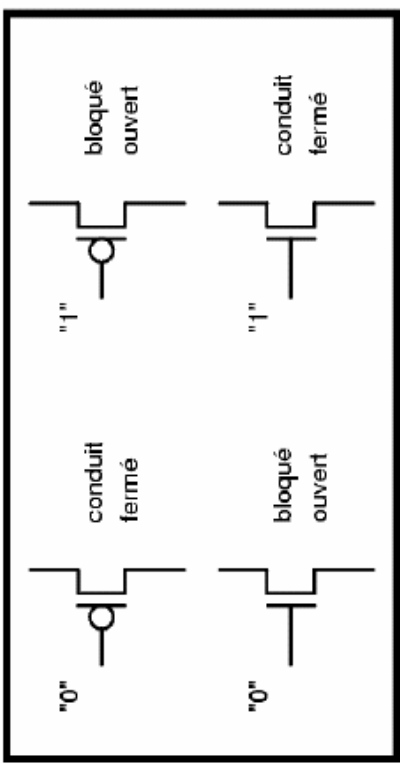


◆ Lecture I²C

I²C, exemple PCF8574 port //



♦ Timing de transmission



45

Interfaces séries

Bus SPI

Synchronous Peripheral Interface

Motorola

SPI, Synchronous Peripheral Interface (Motorola)

- **Bus synchrone**
- **Horloge séparée**
- **Transmission en full-duplex (in, out)**
- **Multi-maître possible**
- **Sélection de l'esclave par lignes séparées**

Environ 1- 4 Mbit/s

SPI, Synchronous Peripheral Interface (Motorola)

MOSI

Master Out, Slave In,

Sortie sur le maître, entrée sur l'esclave

MISO

Master In, Slave Out,

Entrée sur le maître, sortie sur l'esclave

SCK

Serial Clock

Horloge de transmission série, fourni par le maître

SS

Slave Select, 1 par esclave

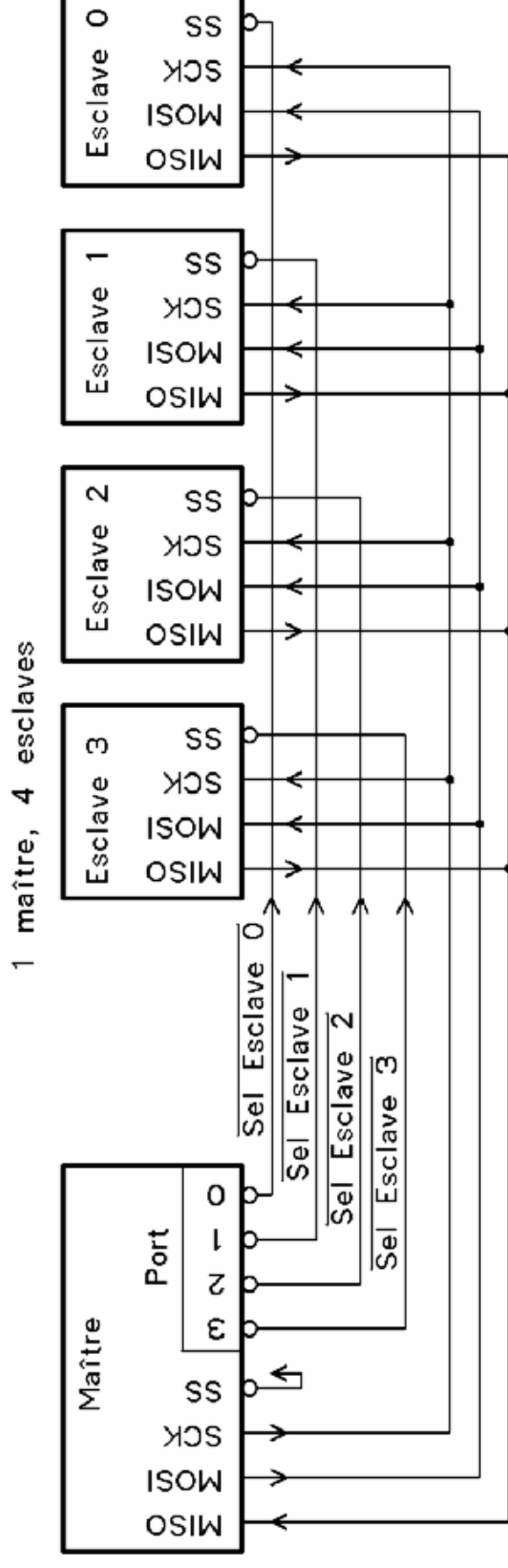
Sélection de l'esclave

Environ 1Mbit/s

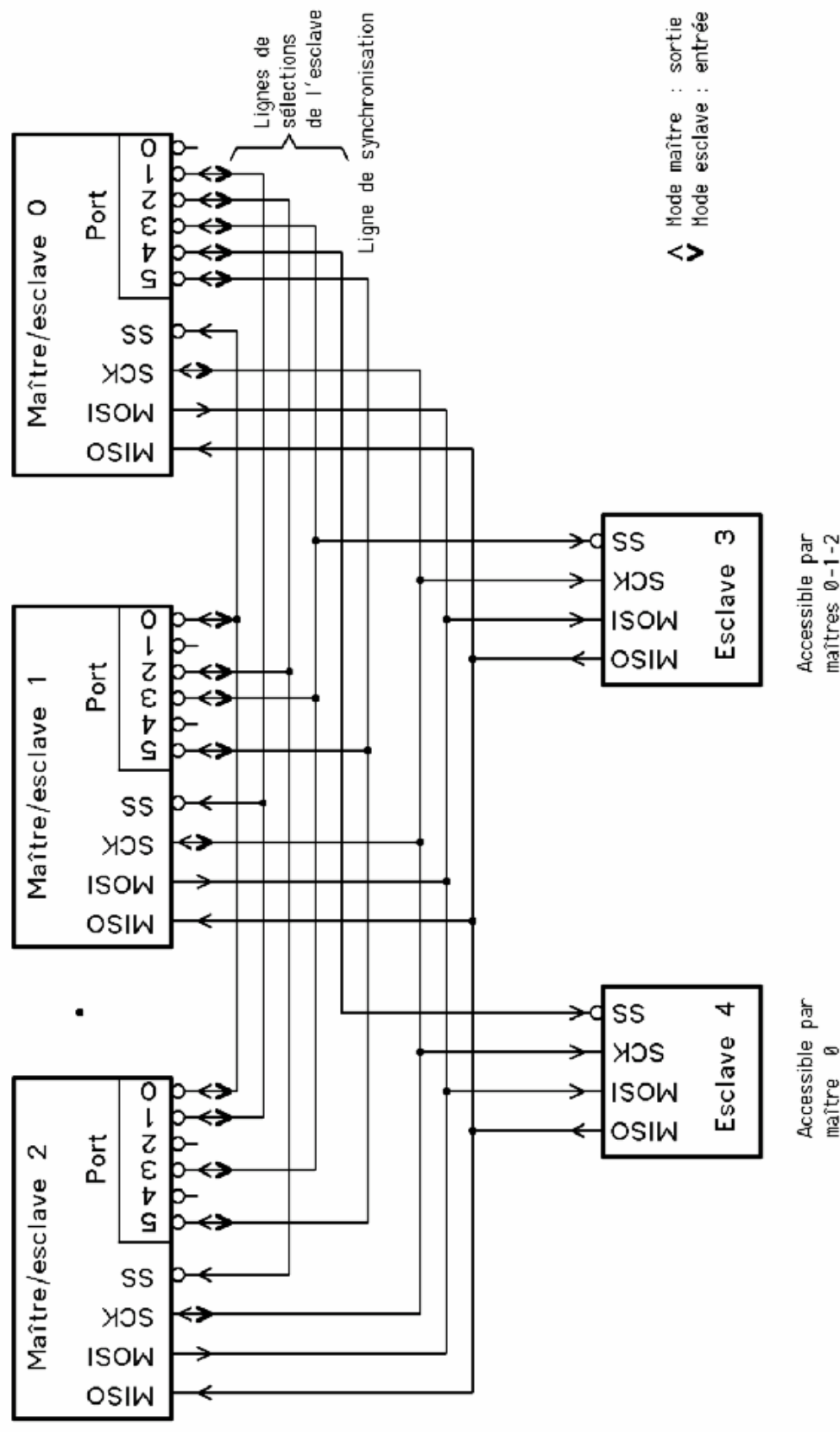
SPI, Synchronous Peripheral Interface (Motorola)

Exemple :

- 1 maître
- 4 esclaves

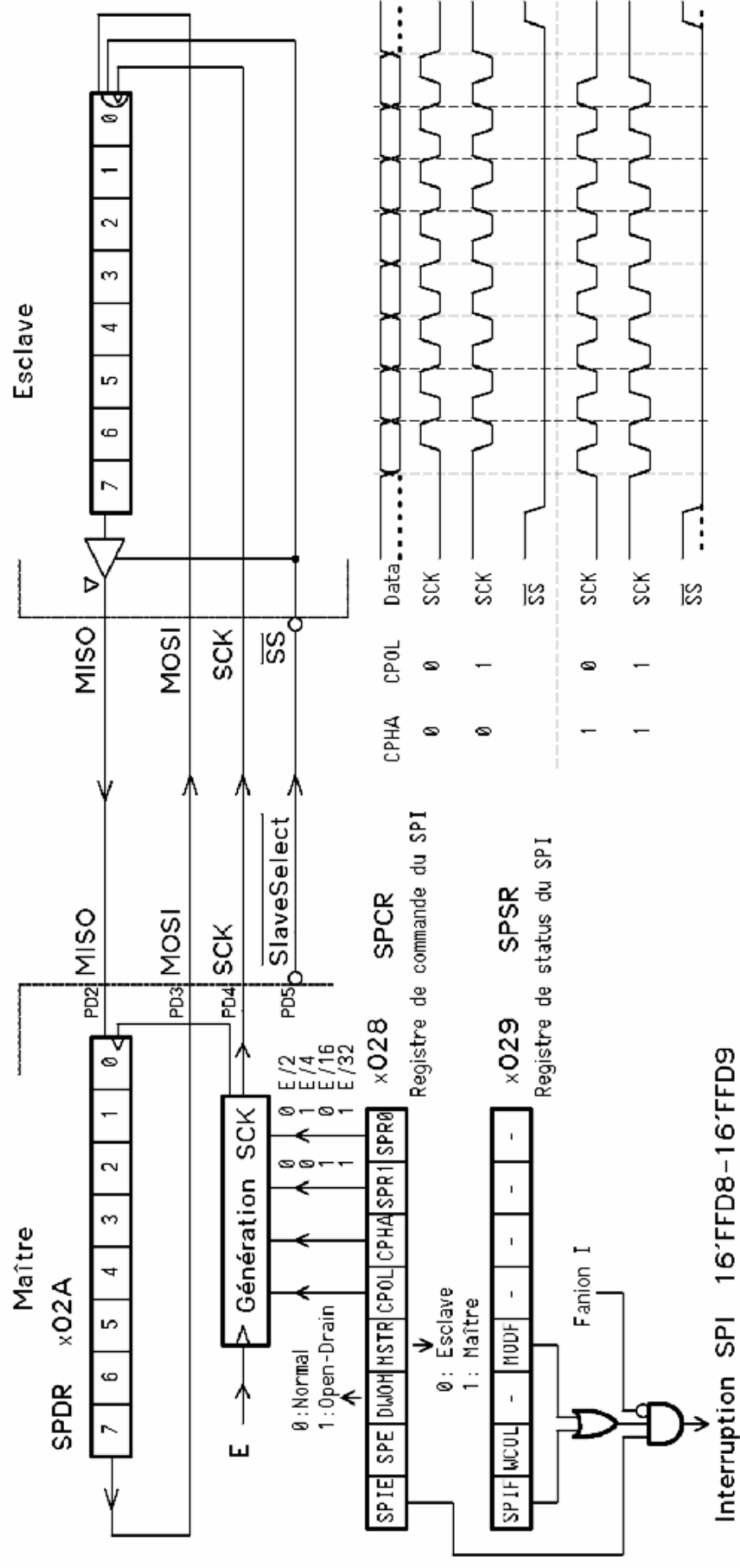


SPI, Synchronous Peripheral Interface (Motorola)



SPI, Synchronous Peripheral Interface (Motorola)

Exemple d'implémentation sur uC 68HC11



Interfaces séries

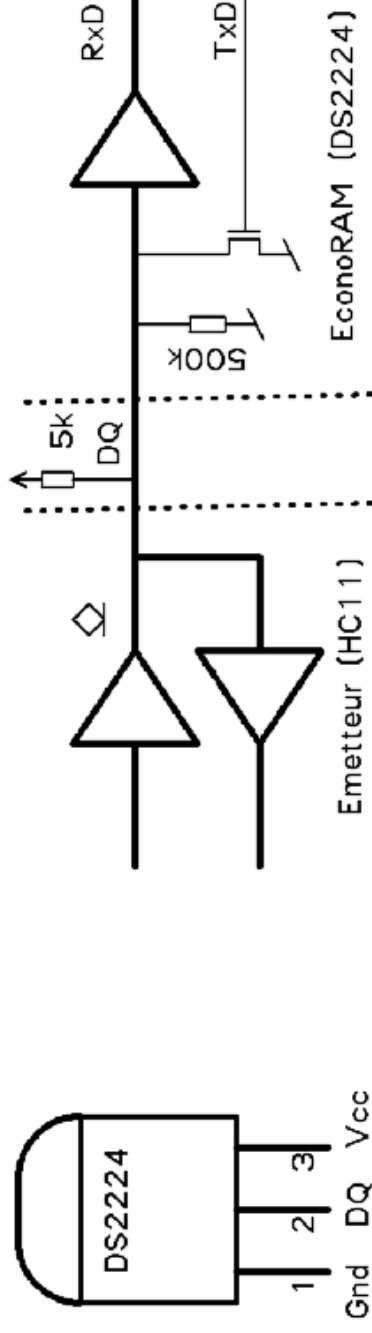
Bus 1-Wire, uLAN

Dallas-Maxim

1-Wire, Dallas/Maxim

- Liaison série un fil en modulation largeur d'impulsion
- **Collecteur ouvert**
- Start bit pour chaque bit, fournit par maître
- ~15 kbit/s
- 50-600m

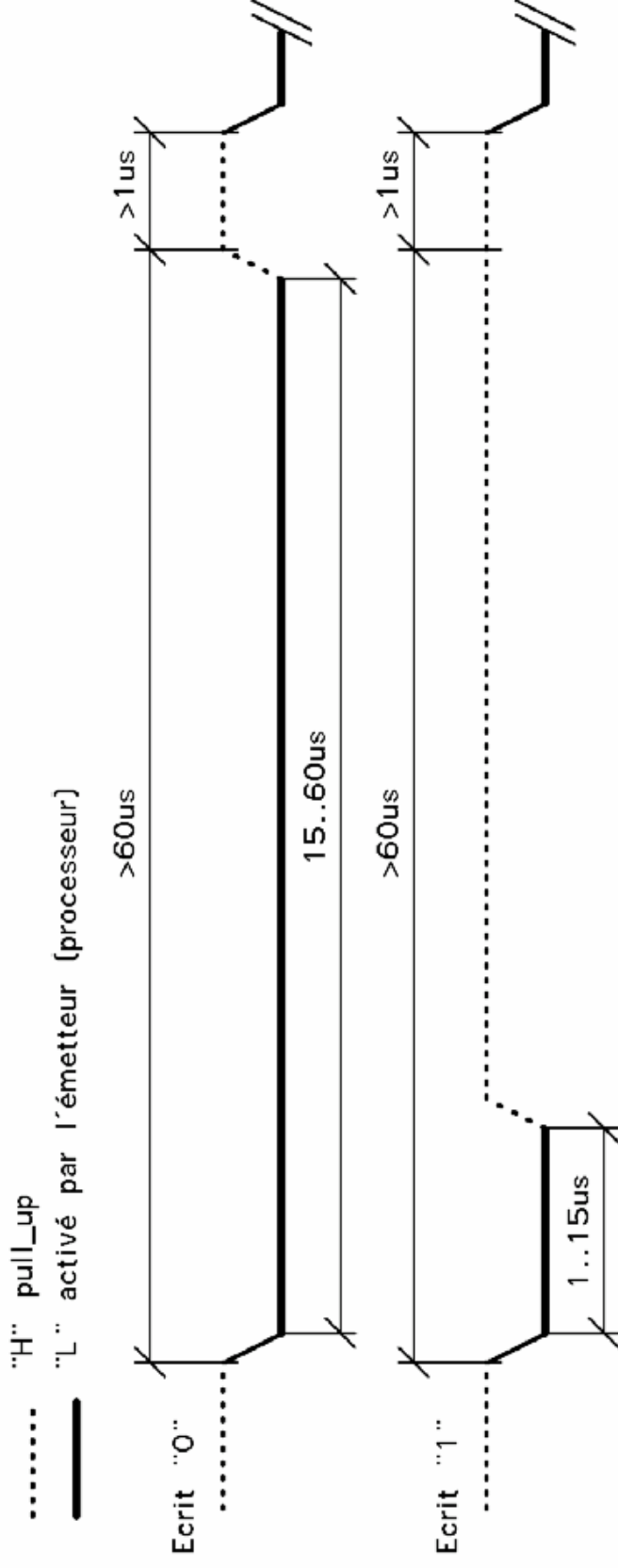
- Exemple de circuit, EconoRAM (ancien circuit)



1-Wire, Dallas

- Liaison série un fil en modulation largeur d'impulsion
- Collecteur ouvert
- Start bit pour chaque bit, fournit par maître

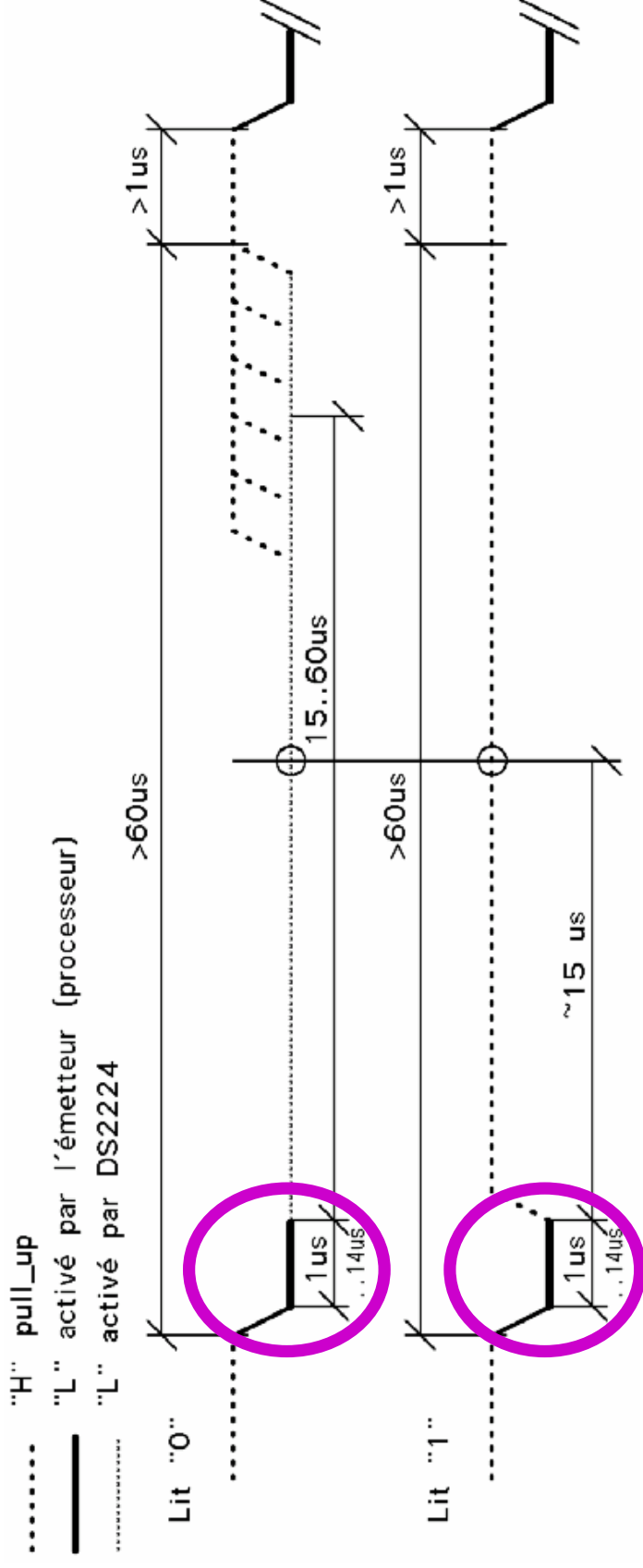
Ecriture



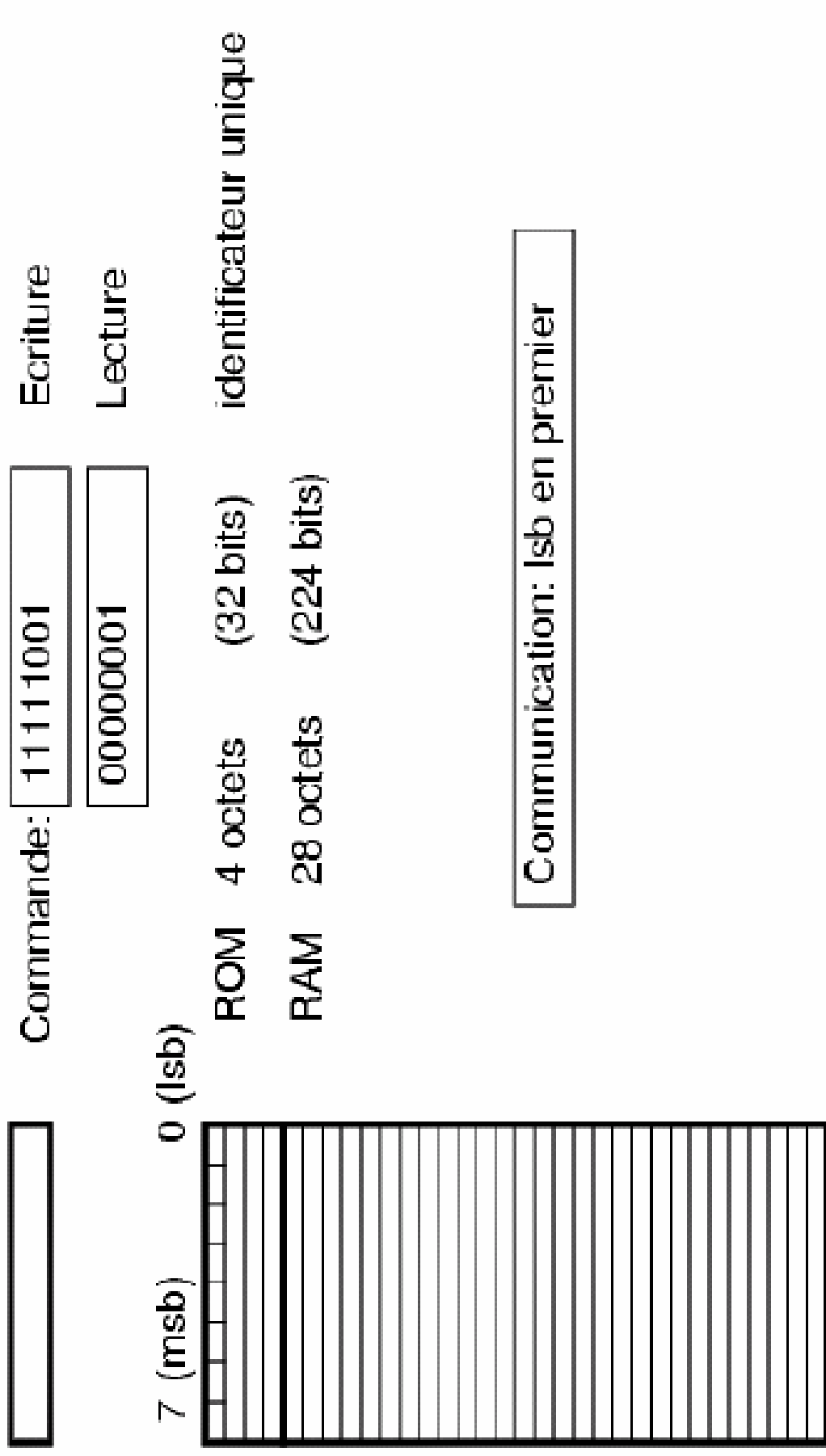
1-Wire, Dallas

- Liaison série un fil en modulation largeur d'impulsion
- Collecteur ouvert
- Start bit pour chaque bit, fournit par maître

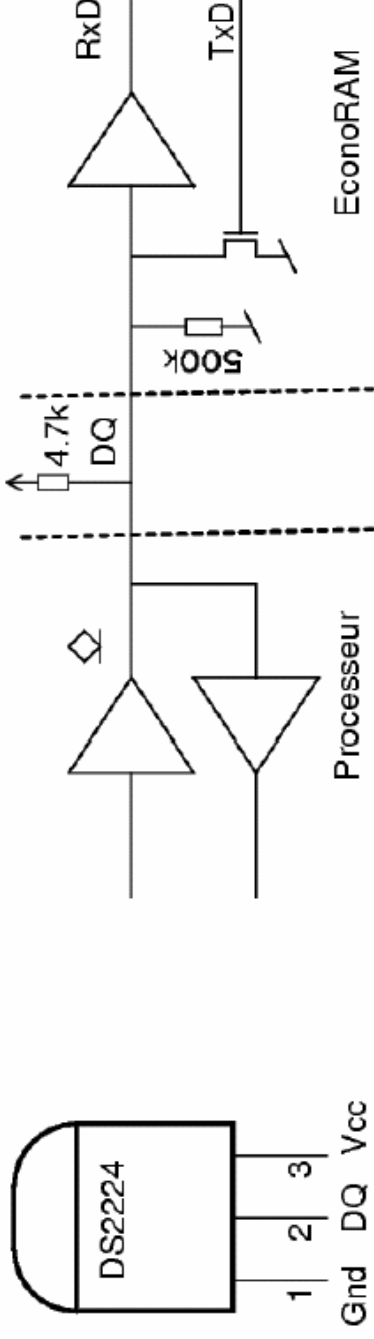
Lecture



Mémoire Dallas 1-Wire



Mémoire Dallas 1-Wire



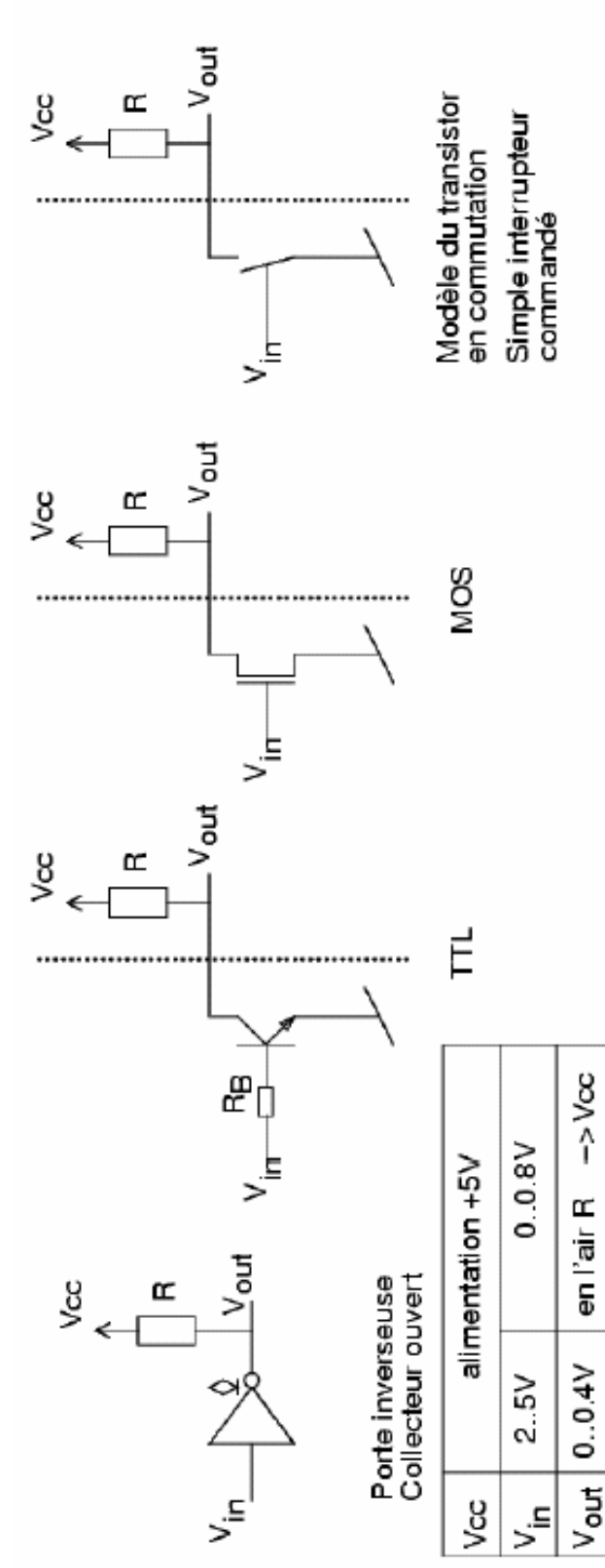
Communication :

- par collecteur ouvert
- synchronisation par bit
- activation start par maître

Interfaces séries

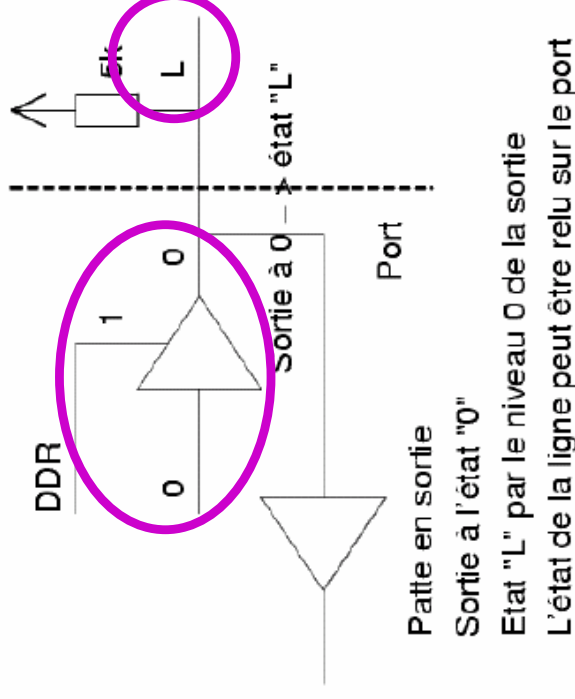
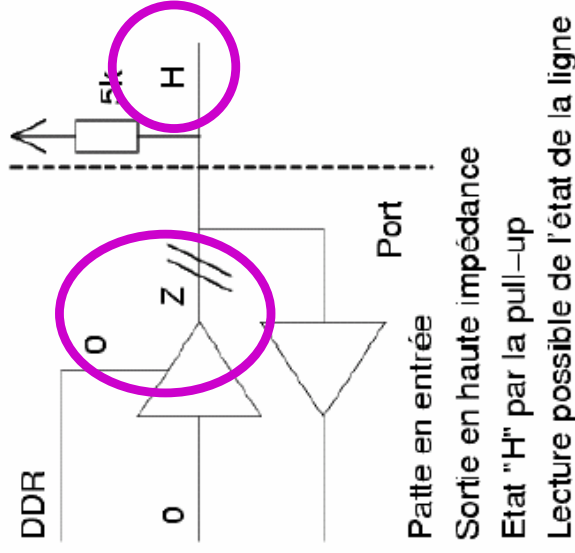
Réalisation de la
fonction collecteur
ouvert avec un port //
programmable

Collecteur ouvert sur port //



◆ Modèle d'un collecteur ouvert

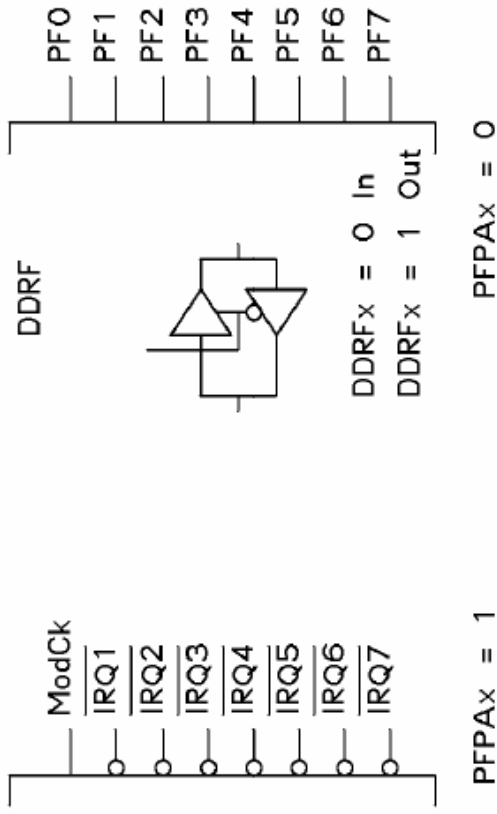
Collecteur ouvert sur port //



- ◆ Modèle d'un collecteur ouvert sur un port d'un interface programmable
- ◆ Utilisation de la direction du bit :
 - entrée → 'Z'
 - Sortie '0' → '0'

Collecteur ouvert sur port // 68331

Reset: DB9 = 1 -> PFPAR = FF DB9 = 0 -> PFPAR = 00

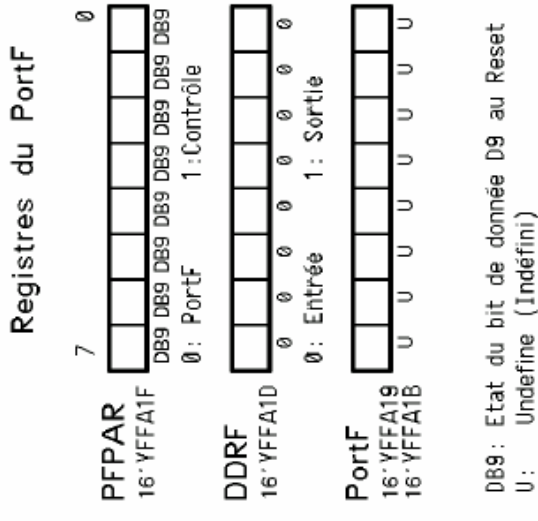


PFPAR : bit x du registre PFPAR (Port F Pin Assignment Register)

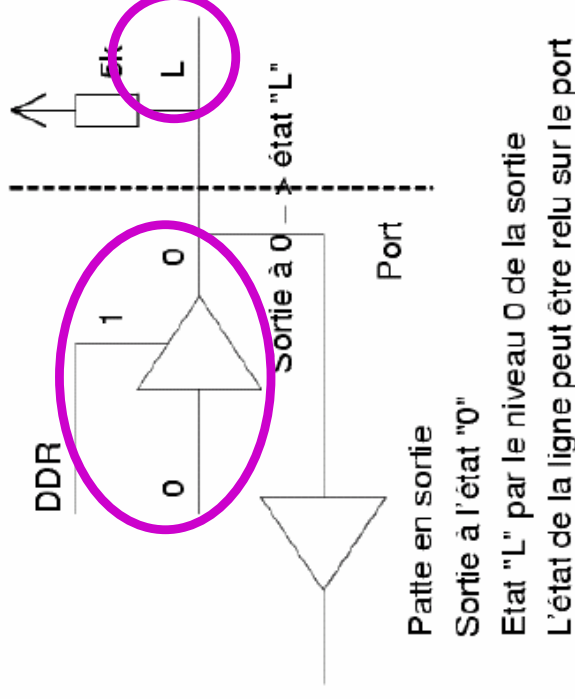
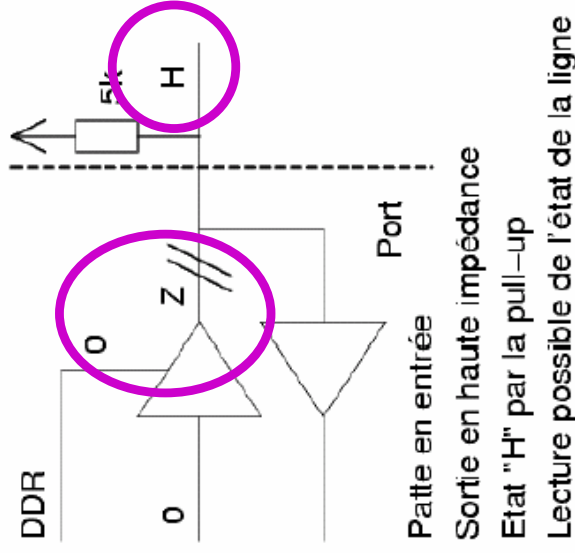
PFPAR : Choix du mode (IRQ \leftrightarrow Port parallèle)

DDRF : Sens du port

PortF : Donnée du port



Collecteur ouvert sur port //



; Macros pour mettre la ligne PortF bit 7 à "0" ou "Z"

IOZ: MACRO

bClr.b #7,DDRF

ENDM

; en entrée → Z