

Le bus CAN (*Controller Area Network*)

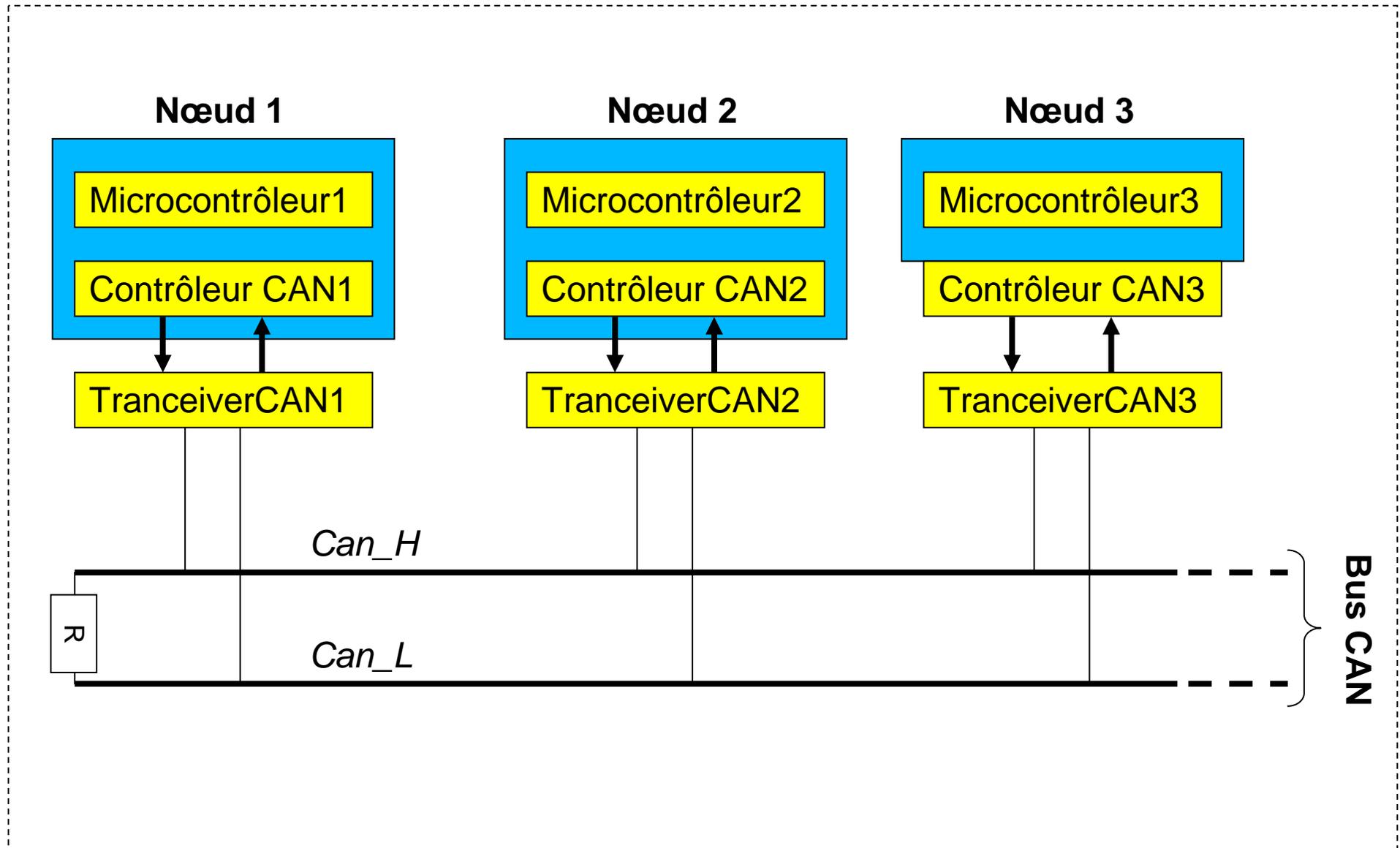
Université de Karlsruhe et Bosch (années 1980..)

- Bus série asynchrone
- Un seul message à la fois
- Tous les nœuds peuvent envoyer des messages
- Tous les nœuds reçoivent les messages
- Les conflits (collisions) et les erreurs sont gérés automatiquement
- Chaque message possède un identifiant
- Un nœud peut filtrer les identifiants pour ne conserver que certains messages
- Couches 1 et 2 du modèle OSI (Physique et liaison de données)

Bibliographie

- <http://www.semiconductors.bosch.de/media/pdf/canliteratur/can2spec.pdf>
- <http://www.can-cia.de/>
- http://fr.wikipedia.org/wiki/Controller_Area_Network
- <http://sitelec.org/download.php?filename=cours/canbus.pdf>

Vue d'ensemble

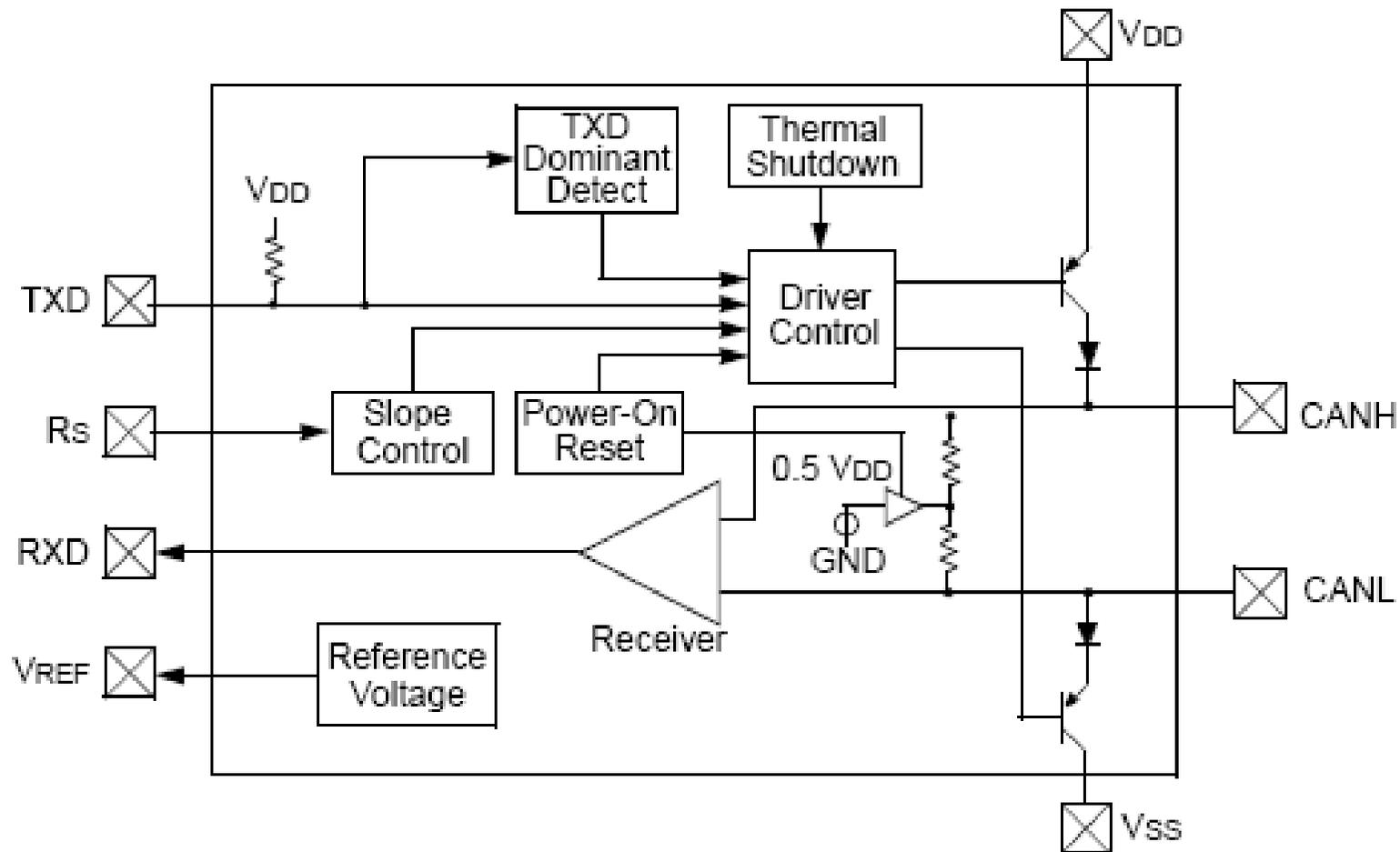


Arbitrage du bus

- Bits dominants (0) – bits récessifs (1)
- Si deux nœuds sont en conflit le bit dominant l'emporte, le nœud dominé arrête l'émission et continue à écouter le bus, il fera une nouvelle tentative lorsque le bus sera libre.
- Le transceiver CAN gère uniquement les niveaux électriques
- Le contrôleur CAN s'occupe de tout le reste...

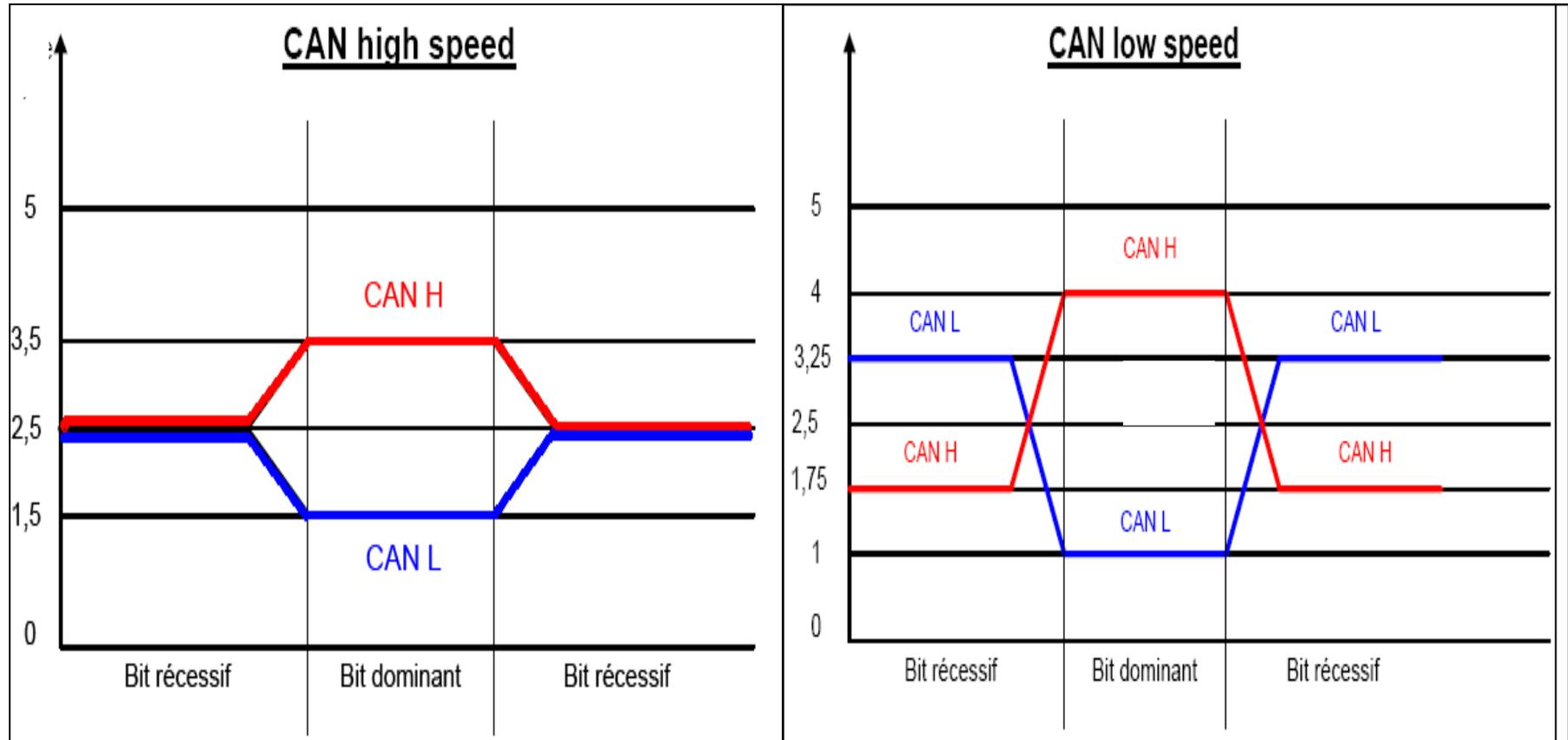
Tranceiver CAN

- Exemple le MCP2551



Niveaux électriques

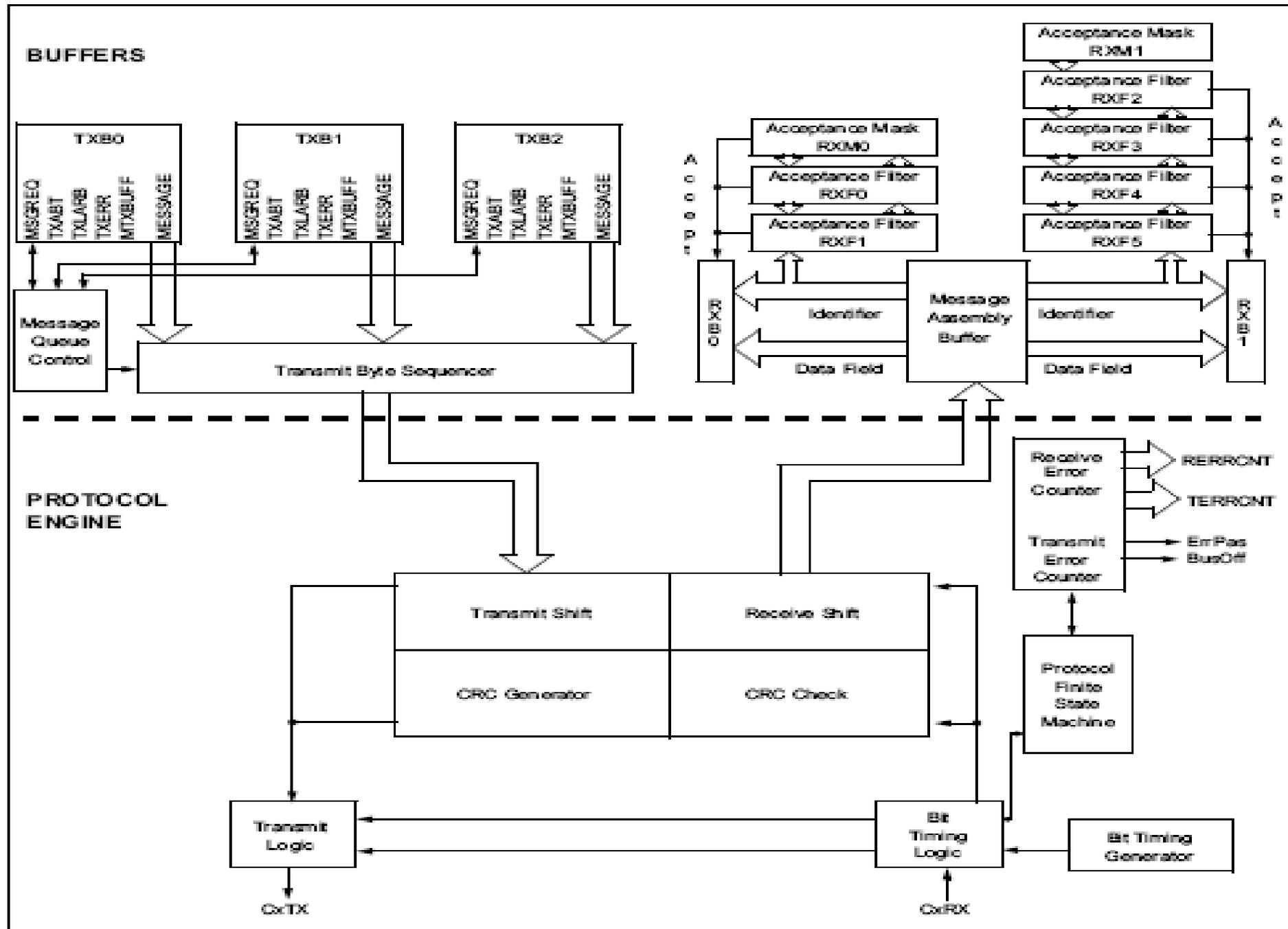
Transmission différentielle



Contrôleur CAN

- Intégré au microcontrôleur (Microchip, TI, freescale, etc...)
- Externe au microcontrôleur (SJA1000...)
- Rôle:
 - Emission et réception des trames, envoi automatique sur demande (request)
 - Mise en trames des données à envoyer
 - Réception, filtrage des trames et récupération des données
 - Acquiescement automatique des trames reçues (ACK)
 - Stockage des données reçues ou à envoyer dans des « FIFO »
 - Gestion sophistiquée des erreurs et de la surcharge du bus...
- Mise en œuvre simple:
 - Registre à initialiser (débit, filtre/masque pour les identifiants...)
 - Possibilité d'utiliser les interruptions...

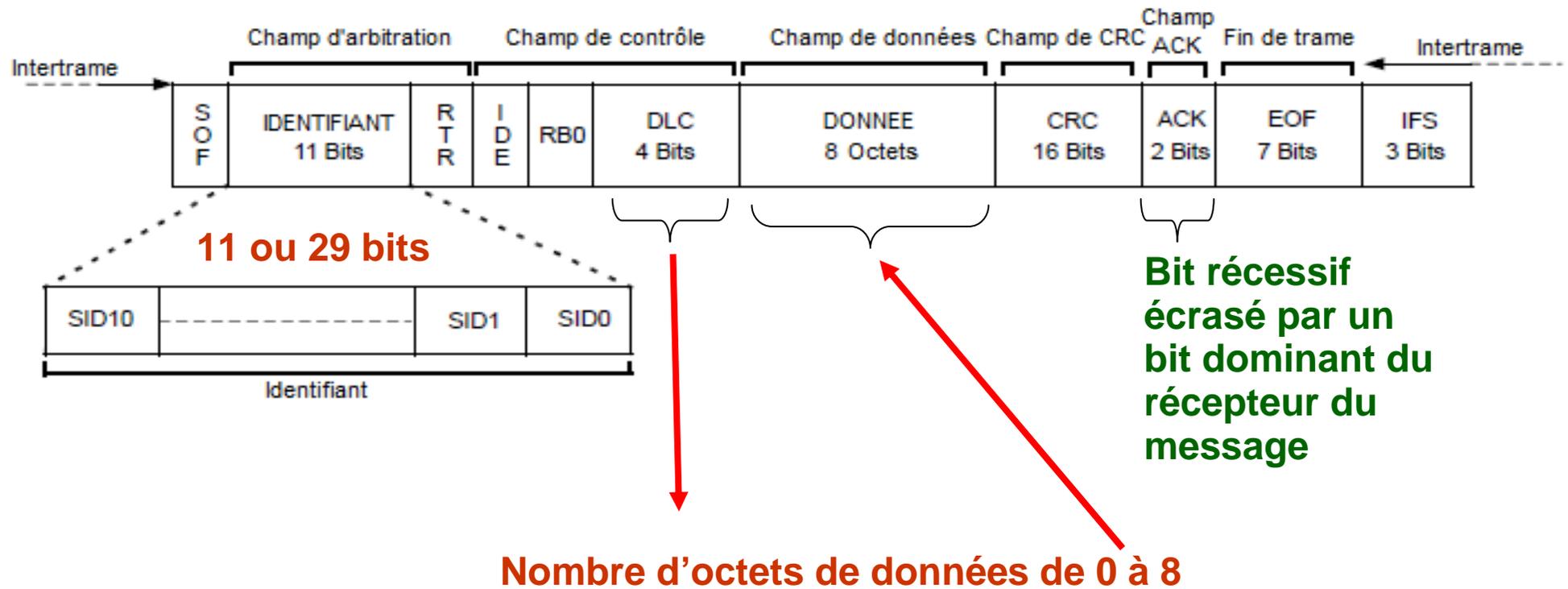
Contrôleur CAN du DSPic30Fxxxx



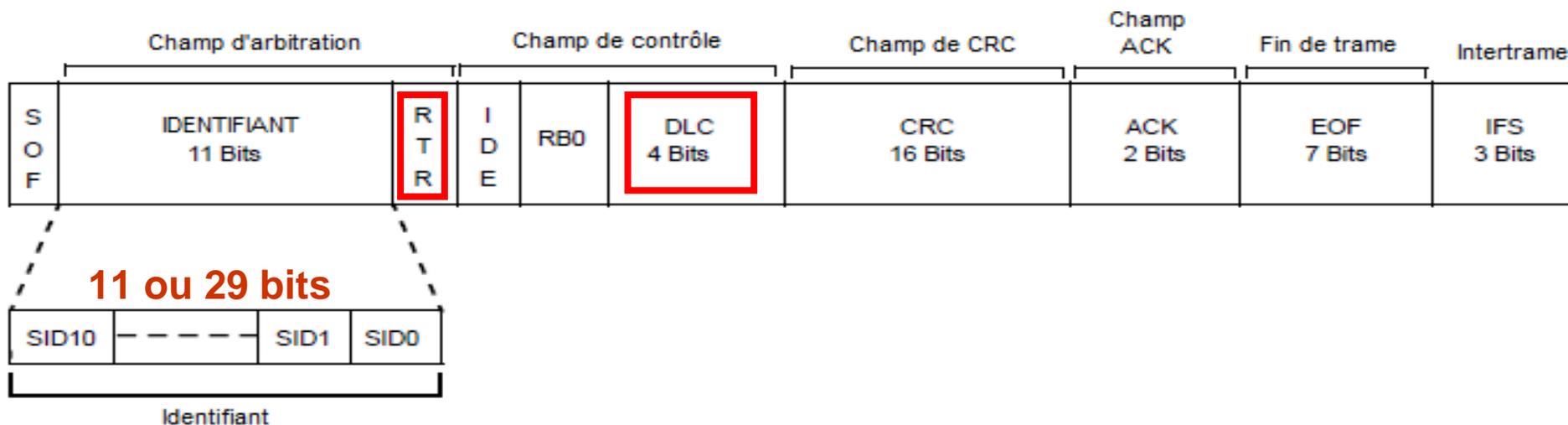
Trame CAN

- Trames de données
- Trames de requête
- Trames d'erreur
- Trames de surcharge

Trame de donnée CAN

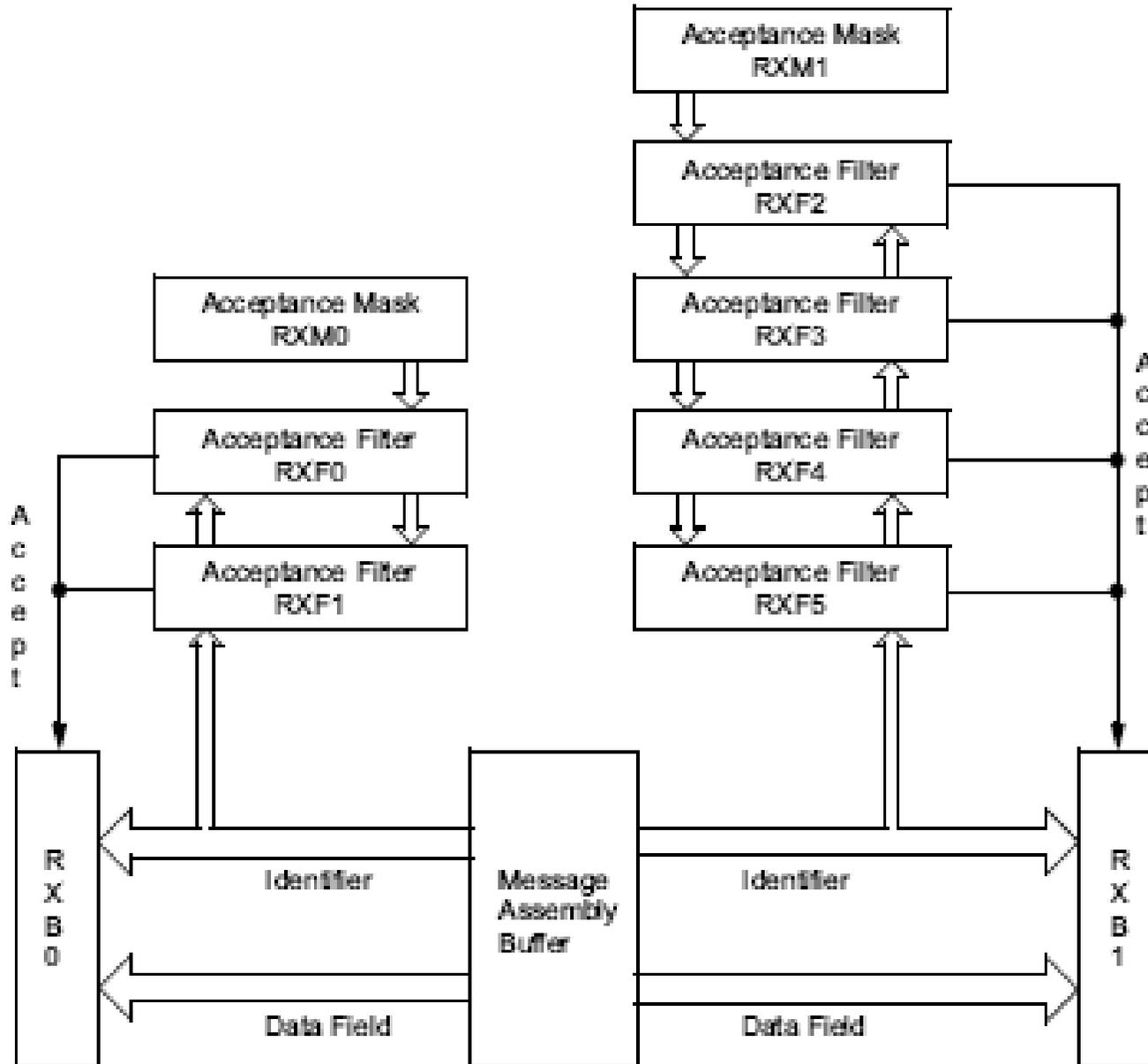


Trame de requête CAN



Lorsque le destinataire correctement configuré accepte cette trame il répond en retournant une trame de même identifiant avec des données.

2 registres de réception



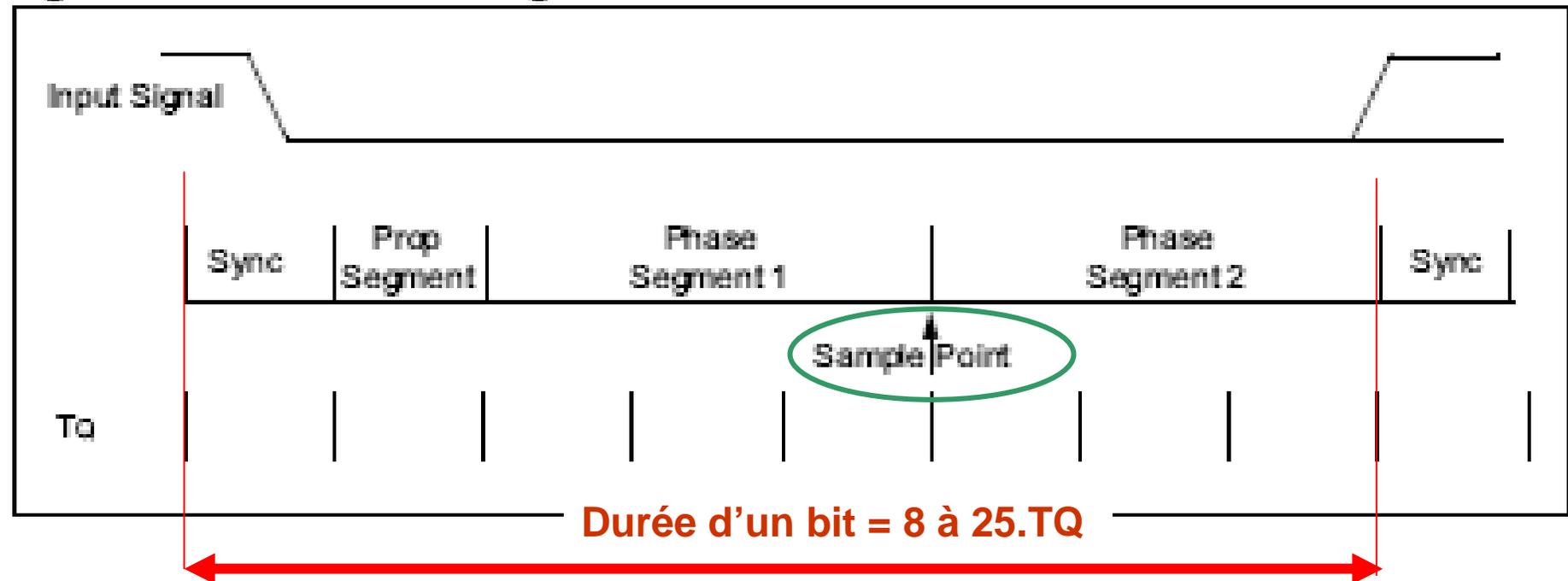
Masque et filtres

- On peut filtrer les messages reçus en fonction de leur identifiant grâce aux registres filtres et masques.
- Filtre:
 - Seuls les bits non masqués sont testés
 - Ils doivent correspondre au bits du filtre pour que le message soit accepté
- Masque:
 - les bits de l'ID correspondants aux « 0 » du masque ne sont pas filtrés (ils sont acceptés)

Configuration « quantum » TQ

The time segments and also the nominal bit time are made up of integer units of time called time quanta or TQ. By definition, the nominal bit time has a minimum of 8 TQ and a maximum of 25 TQ. Also, by definition the minimum nominal bit time is 1 μ sec, corresponding to a maximum 1 MHz bit rate.

Figure 23-21: CAN Bit Timing



Exemple de calcul

Données :

- Débit souhaité $F_{bit} = 1\text{Mbit/s}$
- Quartz : $F_{quartz} = 8\text{MHz}$
- PLL : 8

Calcul:

- $F_{cy} = PLL * F_{quartz} / 4 = 16\text{ MHz}$

Choix :

- $F_{can} = (1 \text{ ou } 4) * F_{cy}$ (configurer C1CTRL)
Soit $F_{can} = F_{cy} = 16\text{MHz}$
- Quantum $F_q = (8 \text{ à } 25) * F_{bit}$ (à choisir...)
Soit 1bit = 8 quantum (configurer C1CFG1-2)
 $\Rightarrow F_q = 8 * F_{bit} = 8\text{MHz}$

Calcul du prédiviseur:

- Prédiviseur de l'horloge quantum BRP :
 $BRP = F_{can} / (2 * F_q) - 1$
- Le calcul donne $BRP=0$

Mailbox

Ensemble de registres pour stocker les trames:

- l'identifiant (3 registres 16 bits)
- la donnée (8 octets) = 4 registres 16bits
- les masques
- les filtres

2 « mailbox » de reception RX0 et RX1

3 « mailbox » d'émission TX0 TX1 et TX2

Interruptions CAN

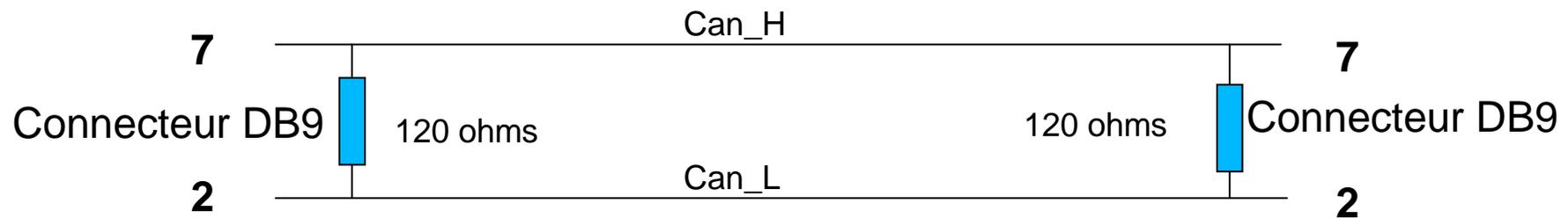
Plusieurs sources :

- message reçu
- émission terminée
- erreur

Si plusieurs sources sont autorisées, la routine de traitement doit tester l'origine de la demande

TP CAN sur DSP 30f4011

- Télécharger et étudier le programme de test
- Ce programme se pilote par le port série (38400bds,N,8,1)
- Relier deux cartes de TP (quartz 8 MHz) par le bus CAN



TP CAN sur DSP 30f4011

- **Vérifier le bon fonctionnement**
- **Observer les trames à l'oscilloscope sur TXcan, RXcan, CanH et CanL**
- **Compléter ce programme pour:**
 - Choisir l'identifiant du message à envoyer
 - Recevoir uniquement les messages d'identifiants compris uniquement entre 0x2000 et 0x2FFF sur RX0 et les messages 0x3000 et 0x4000 sur RX1
 - Changer la vitesse entre 1Mbit/s et 125kbits/s
 - Utiliser le mode requête
 - Gérer toutes les sources d'interruptions CAN