



I2C

Introduction au bus I2C

Camille Diou

Docteur en microélectronique



LABORATOIRE INTERFACES
CAPTEURS & MICROÉLECTRONIQUE



UNIVERSITÉ
DE METZ

Introduction au bus I2C

Présentation du bus I2C

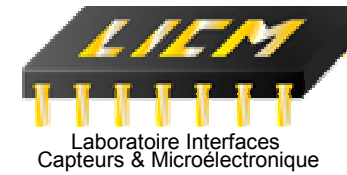
Le protocole I2C

Les conflits

Spécificités et évolutions



I2C



Le bus I2C

Camille Diou

Docteur en microélectronique



I2C

Introduction

Présentation du bus I2C



Présentation



I2C : Inter Integrated Circuit

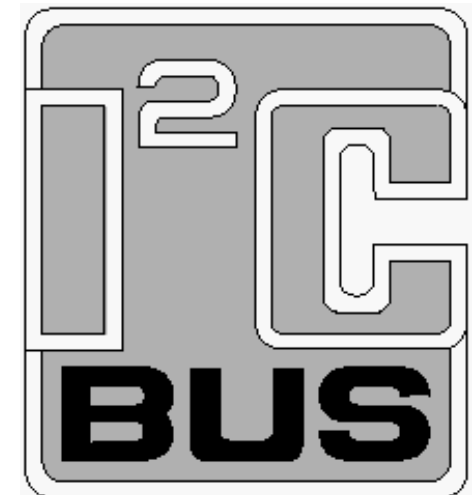
développé au début des années 80 par Philips Semiconductor pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur moderne.



But :







faire communiquer entre eux des composants électronique très divers grâce à seulement 3 fils :

- Signal de donnée : SDA
- Signal d'horloge : SCL
- Signal de référence électrique : masse





Caractéristiques du bus I2C

-  Deux lignes uniquement (SDA et SCL) + masse
-  1 adresse unique pour chaque périphérique
-  Bus multi-maître, détection des collisions et arbitrage
-  Bus série, 8 bits, bi-directionnel à 100 kbps (*standard mode*), 400 kbps (*fast mode*), 3,2 Mbps (*high-speed mode*)
-  Filtrage intégré : réjection des pics parasites
-  Nombre de circuits uniquement limité par la capacitance maximale du bus : 400 pF

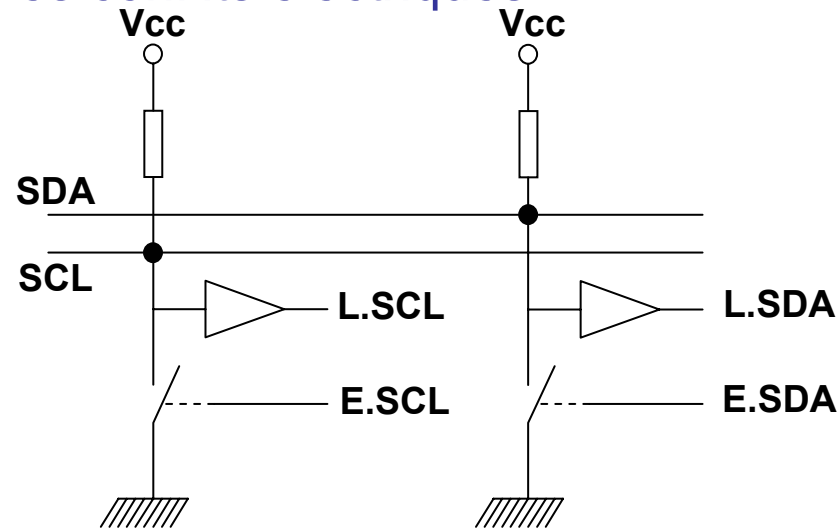


Principe du bus I2C



Entrées-sorties de type collecteur ouvert

rôle : éviter les conflits électriques



Les lignes SDA et SCL peuvent uniquement être forcée au niveau bas.
On ne peut pas les forcer au niveau haut.



I2C

Protocole

Gestion du bus I2C




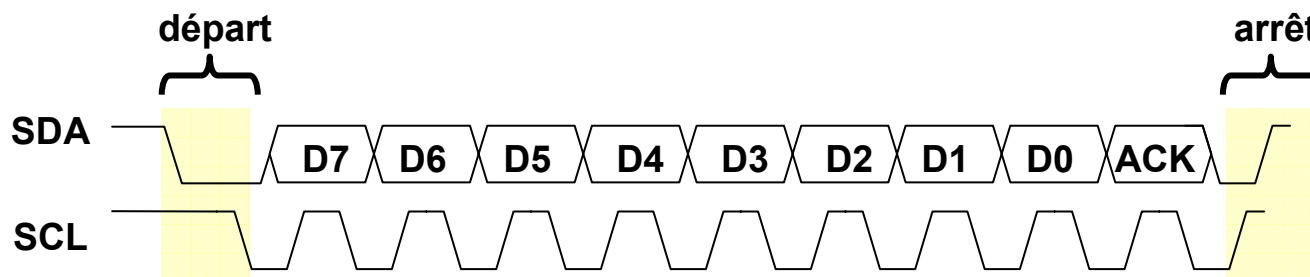
Prise de contrôle du bus

 Le bus doit être au repos avant la prise de contrôle SDA et SCL à 1

 Pour transmettre des données, il faut surveiller :

- La condition de départ : SDA passe à 0, SCL reste à 1
- La condition d'arrêt : SDA passe à 1, SCL reste à 1

 Après avoir vérifié que le bus est libre, puis pris le contrôle de celui-ci, le circuit en devient le maître : c'est lui qui génère le signal d'horloge

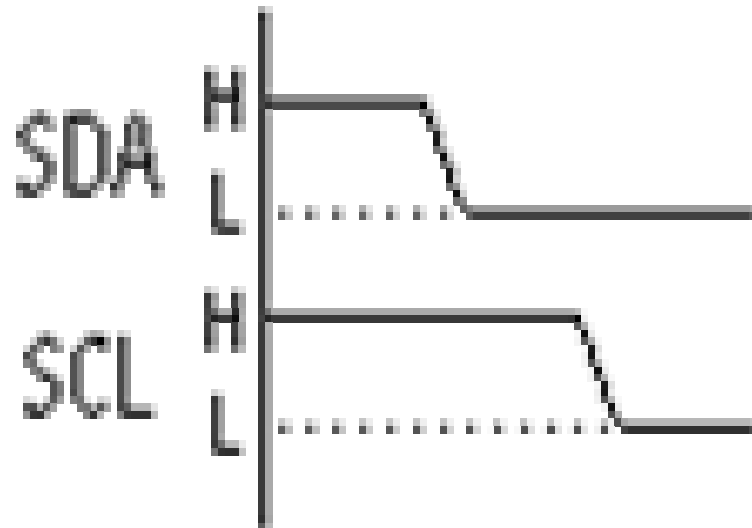




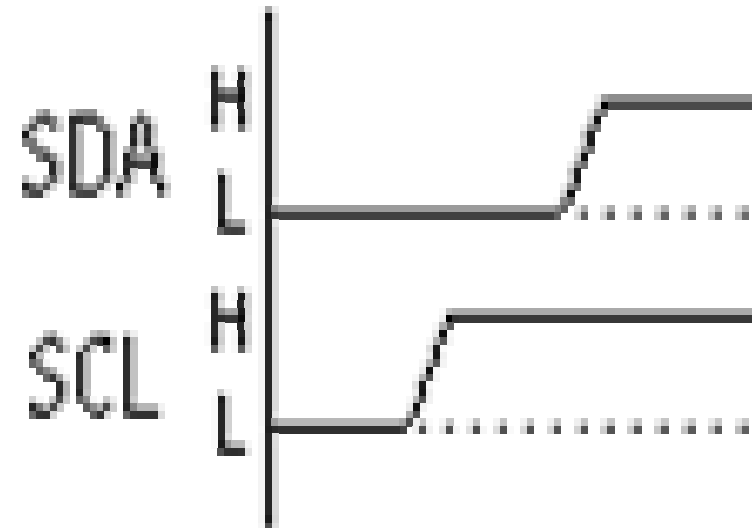
Prise de contrôle du bus



Bit Start









Bit Stop



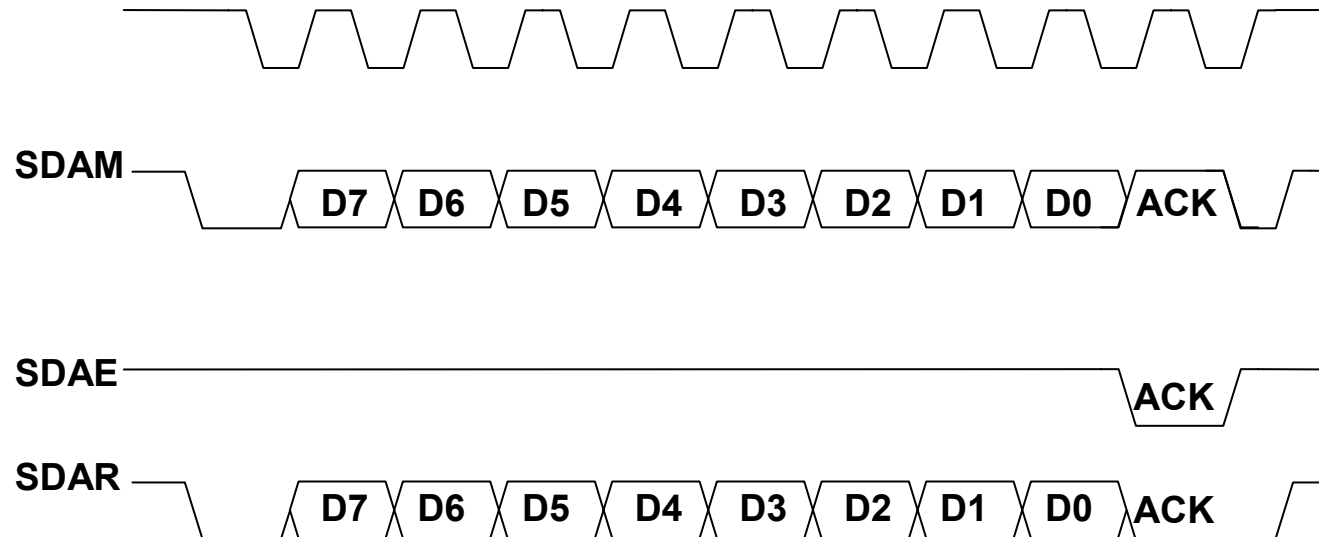


Transmission d'un octet

-  Le maître transmet le bit de poids fort D7 sur SDA
-  Il valide la donnée en appliquant un niveau '1' sur SCL
-  Lorsque SCL retombe à '0', il poursuit avec D6, etc. jusqu'à ce que l'octet complet soit transmis
-  Il envoie le bit ACK à '1' en scrutant l'état réel de SDA
-  L'esclave doit imposer un niveau '0' pour signaler que la transmission s'est déroulée correctement
-  Le maître voit le '0' (collecteur ouvert) et peut passer à la suite



Transmission d'un octet



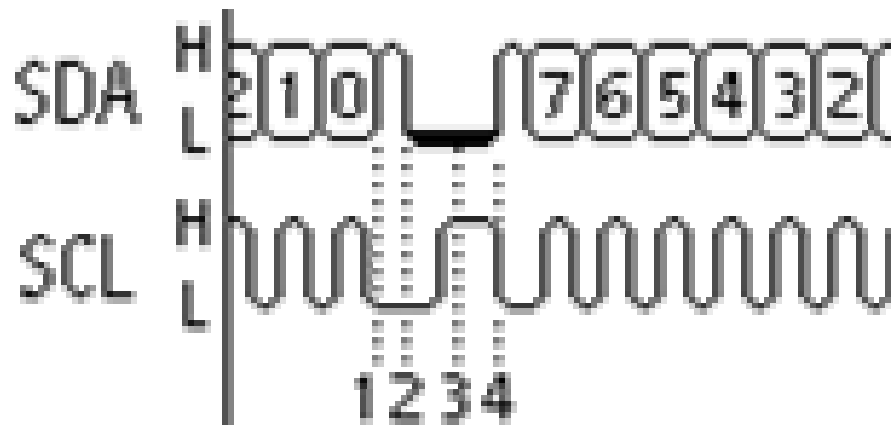
- SCL : Horloge imposée par le maître
- SDAM : Niveaux de SDA imposés par le maître
- SDAE : Niveaux de SDA imposés par l'esclave
- SDAR : Niveaux de SDA réels résultants



Transmission d'un octet



L'acquittement : ACKNOWLEDGE



Le maître libère la ligne SDA



L'esclave force la ligne SDA au niveau bas (trait gras)



Le maître envoie une impulsion sur l'horloge SCL



Lorsque l'impulsion retombe à zéro, l'esclave libère SDA



Transmission d'une adresse



Nombre de composants important : nécessité de définir pour chacun une adresse unique



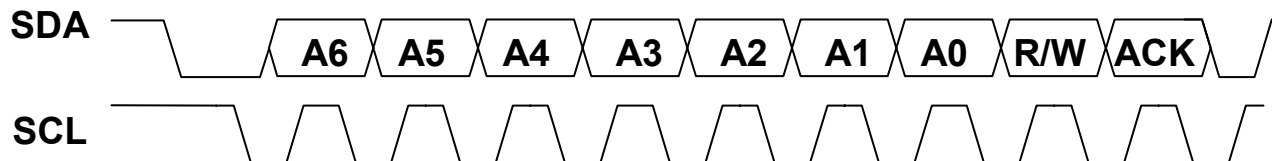
Adresse codée sur 7 bits, définie par :

- Son type
- L'état appliqué à un certain nombre de ces broches



Adresse transmise sous la forme d'un octet au format particulier :

- D7 à D1 : 7 bits d'adresse A6 à A0
- D0 : bit R/W qui détermine si le maître veut lire ou écrire





Transmission d'une adresse



Cas particulier : les mémoires

- Espace adressable plus grand que les autres circuits : adresses codées sur 2 octets ou plus :
 - Premier octet : adresse du circuit
 - Octets suivants : adresse interne de la mémoire



Adresses réservées

- 00000XXX et 111111XX réservées à des modes de fonctionnement particuliers



Mode d'accès à l'esclave



Demande de lecture ou d'écriture ?

- Le 8ème bit d'adresse spécifie le mode d'accès : R/W
- Le nombre maximal de circuits est de 128
- Les adresses PAIRES sont des adresses d'ÉCRITURE, et les adresses IMPAIRES sont des adresses de LECTURE.

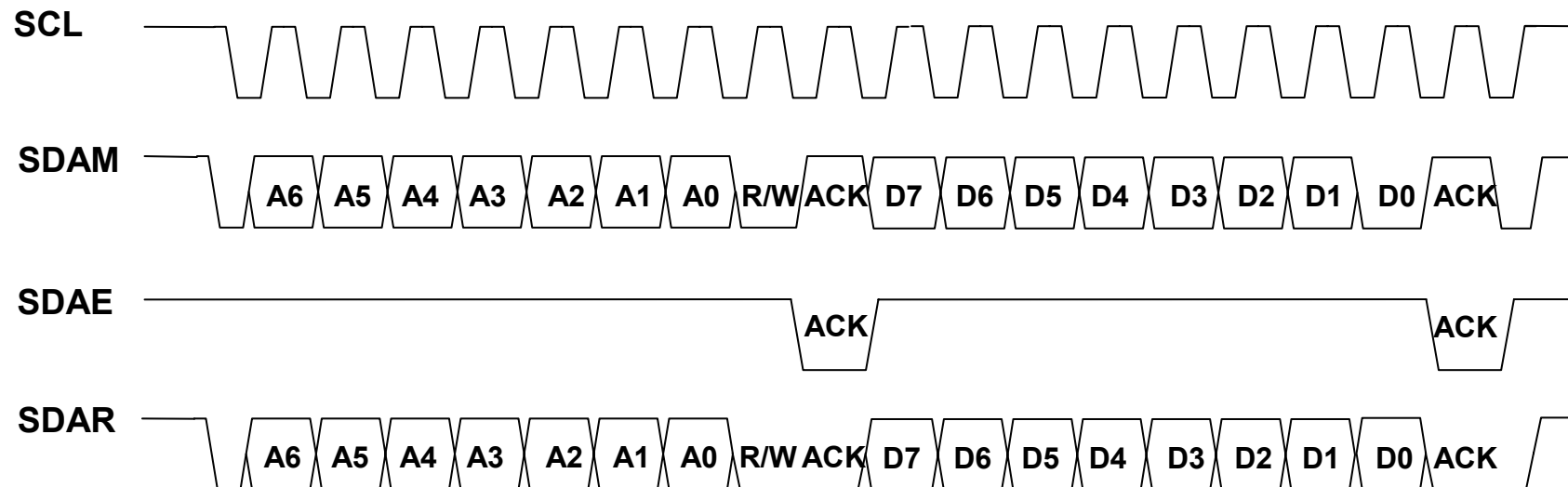
- Exemple : PCF8574 (General purpose 8 BIT I/O port)
 - Adresse esclave d'écriture : $(01000000)_b = 64d$
 - Adresse esclave de lecture : $(01000001)_b = 65d$



Écriture d'une donnée

1. Envoi de l'adresse de destination
2. Sélection du mode écriture (R/W à 0)
3. Envoi de la donnée

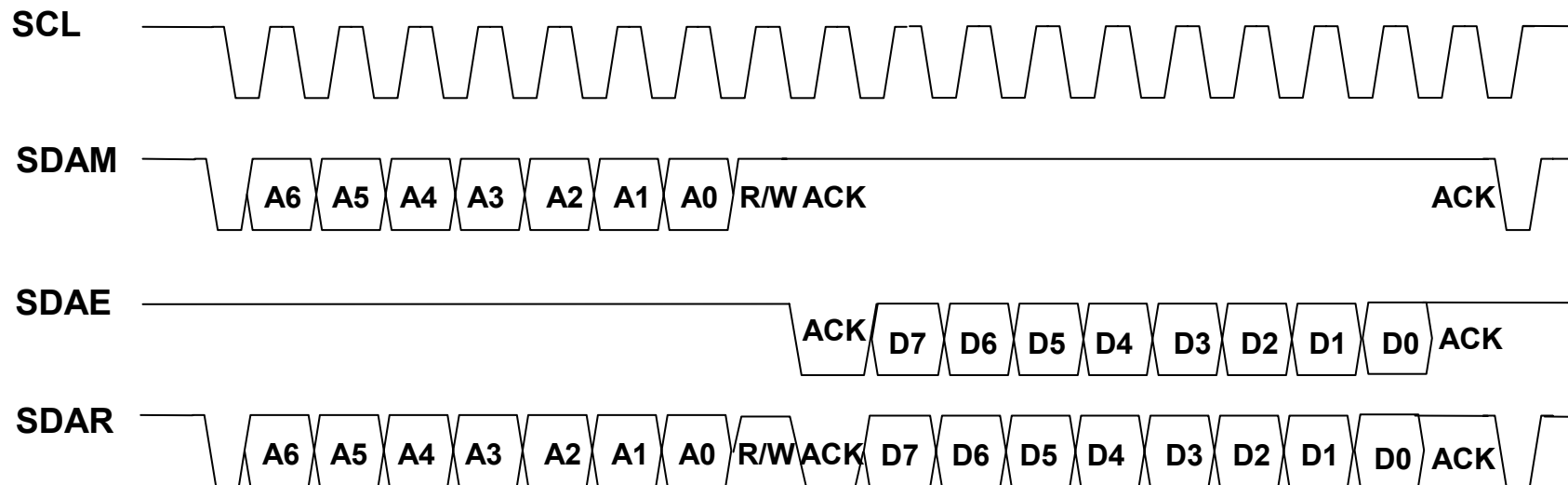
Note : il peut être nécessaire d'attendre ACK avant de poursuivre (écriture dans des mémoire, etc.)





Lecture d'une donnée

1. Le maître envoie l'adresse puis attend l'ACK
2. L'ACK est positionné par l'esclave, puis celui-ci émet les données sur SDA
3. Ensuite, le maître positionne ACK à '0' pour continuer la lecture, ou '1' pour stopper la transmission





I2C

Conflits

Que le meilleur gagne !



Problème



Le bus I2C est de conception destiné à accueillir plusieurs maîtres \Rightarrow problème commun à tous les réseaux utilisant un canal de transmission unique : comment arbitrer ?



Chaque maître peut prendre possession du bus dès que celui-ci est libre : possibilité que deux maîtres prennent la parole en même temps



Pas de problème électrique \Rightarrow collecteur ouvert



Problème logique \Rightarrow éviter la corruption des données due à la collision des bits transmis



Principe

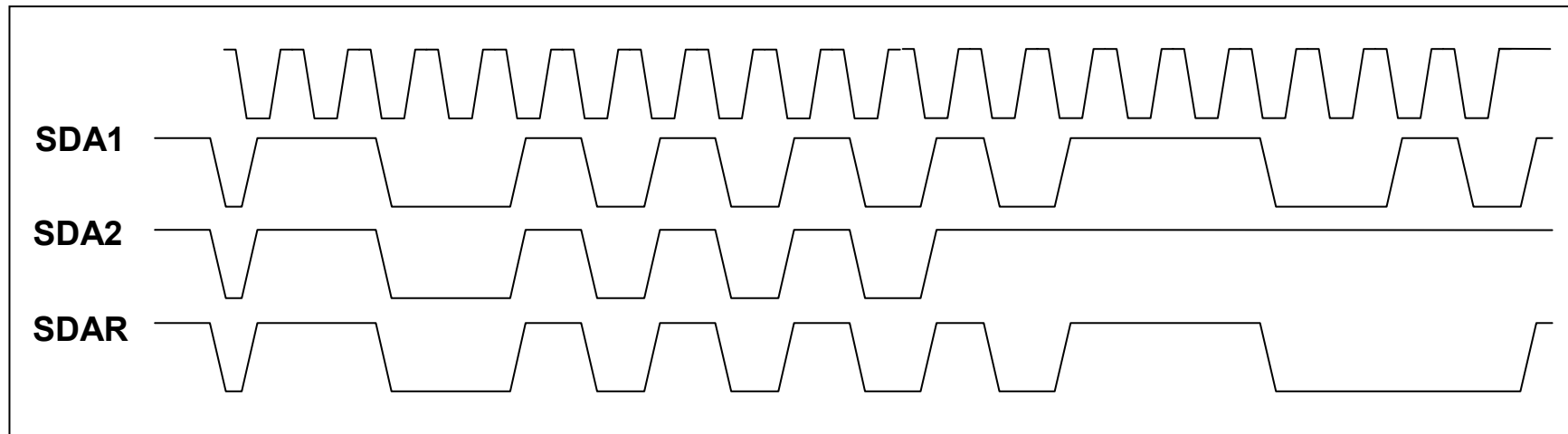


Prise de contrôle du bus

- Vérifier que le bus est libre
- Condition d'arrêt envoyée depuis au moins 4,7 μ s
- Prise de contrôle effectif, mais vérification de l'état des lignes SDA et SCL. Plusieurs cas :
 - Différents maîtres envoient les mêmes données en même temps : aucun conflit, cas rare
 - Un maître impose un '0' : il relire obligatoirement un '0' et continuera à transmettre
 - Un maître cherche à appliquer un '1' sur le bus
 - S'il lit '1', il continue à transmettre
 - S'il lit '0', un autre maître a pris la parole en même temps : il perd l'arbitrage, arrête d'émettre, mais continue à lire



Exemple



SCLR : Horloge résultante

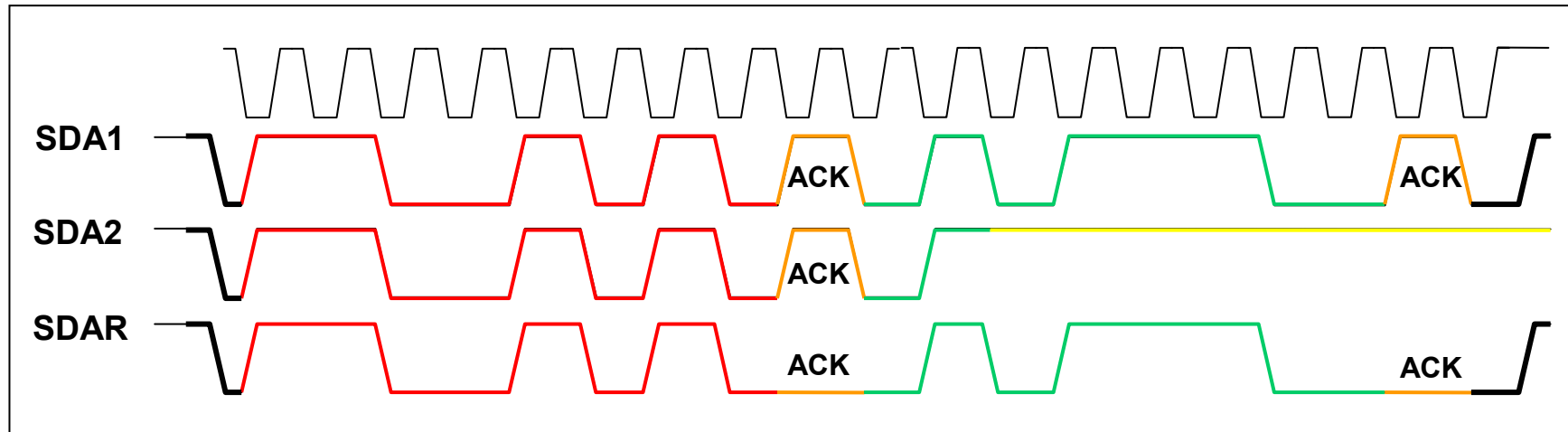
SDA1 : Niveaux de SDA imposés par le maître n°1

SDA2 : Niveaux de SDA imposés par le maître n°2

SDAR : Niveaux de SDA réels résultants lus par les deux maîtres



Exemple



Premier octet transmis normalement : les 2 maîtres imposent les mêmes données

- Deuxième octet : n°2 impose '1' mais relit '0' ; il perd le contrôle du bus et devient esclave



Le maître n°1 ne voit pas le conflit et continue d'émettre normalement, et l'esclave reçoit les données sans erreurs



Synchronisation



Problème : si un périphérique lent est connecté sur le réseau



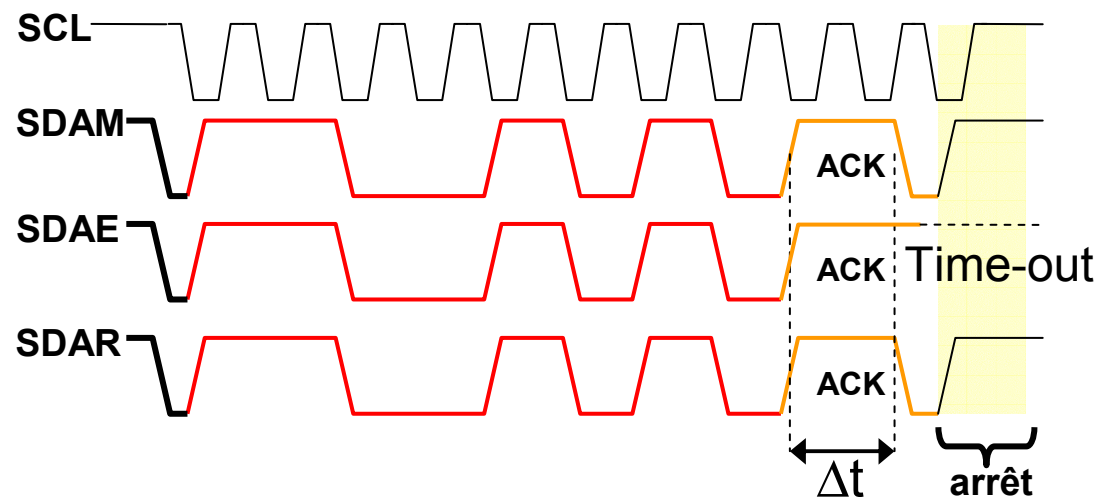
Fonctionnement normal de l'acquittement :

- L'esclave force la ligne au niveau bas immédiatement après le 8ème bit, avant l'impulsion d'acquittement émise par le maître.



Exemple d'un CAN (convertisseur analogique numérique) :

- Détection de son adresse
- Démarrage de la conversion
- Attente de la fin de conversion puis acquittement après Δt
- Le maître a considéré que l'esclave n'était pas présent (réponse après Δt)





Synchronisation



Solution : utilisation de la ligne SCL pour conserver la main

- Après avoir mis SDA au niveau bas, l'esclave doit le relâcher dès que le maître relâche SCL
- Si l'esclave force SCL au niveau bas, le maître ne peut plus émettre d'impulsion d'horloge puisqu'il ne peut plus remettre SCL au niveau haut

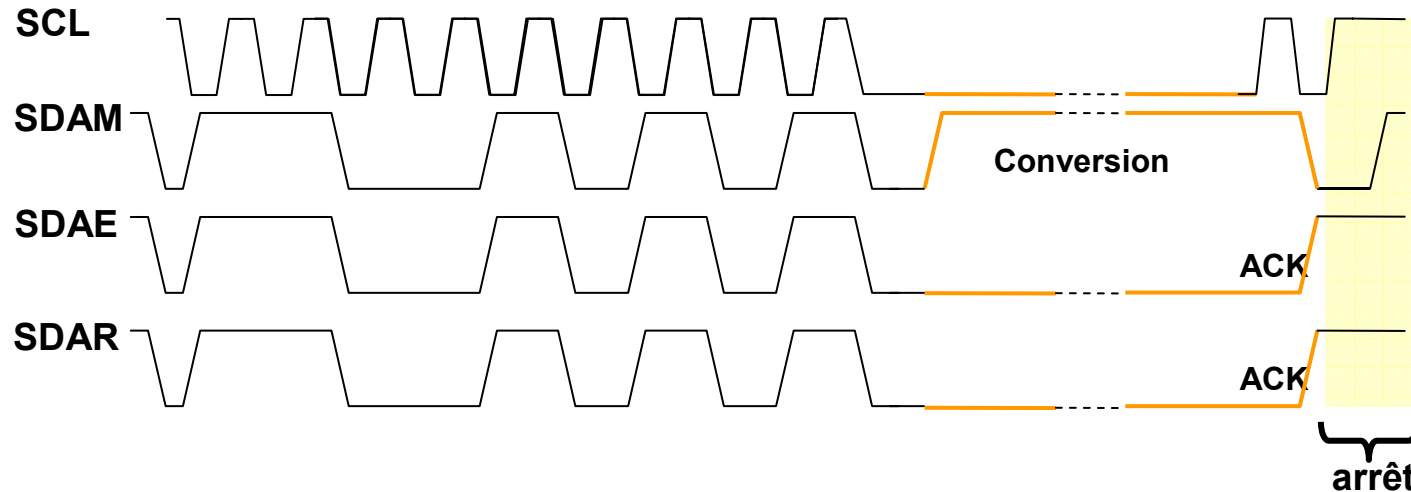


Méthode :

- après avoir reçu le 8ème bit, l'esclave force SDA au niveau bas (acquiescement). En même temps, il démarre la conversion et force SCL au niveau bas également.
- Le maître démarre la phase d'attente d'acquiescement : il relâche SDA puis relâche SCL : lorsqu'il teste le niveau de SCL, il constate qu'il est toujours bas, forcé par l'esclave. Il entre alors dans une boucle en attendant que SCL repasse au niveau haut.
- Dès que le maître détecte que SCL est au niveau haut, il teste SDA : il trouve un niveau bas (acquiescement valide). Il peut alors lire le résultat de la conversion.



Synchronisation



Technique utilisée par les mémoires EEPROMs I2C



Paralysie du bus pendant SCL='0' : si le temps de calcul est long, la perte de temps influe sur les performances de manière importante



En cas de problème électrique, le bus peut se trouver bloqué



Synchronisation



Seconde méthode de synchronisation :

- Premier échange pour la demande de conversion, second échange pour la lecture du résultat :
 - Le maître envoie un message avec l'adresse du CAN
 - L'esclave (CAN) émet l'acquittement et démarre la nouvelle conversion
 - Le maître lit un octet et envoie le bit de stop : résultat de la conversion
- Entre les deux accès, le bus est libre pour un autre échange.
 - Si le convertisseur n'a pas fini la conversion, il n'émet pas de bit ACK. Le maître doit alors retenter plus tard une nouvelle lecture

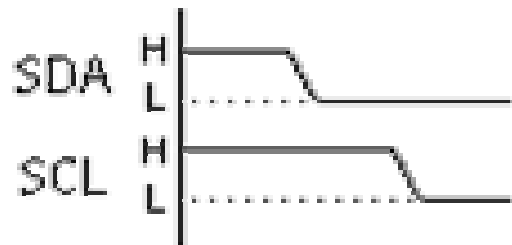
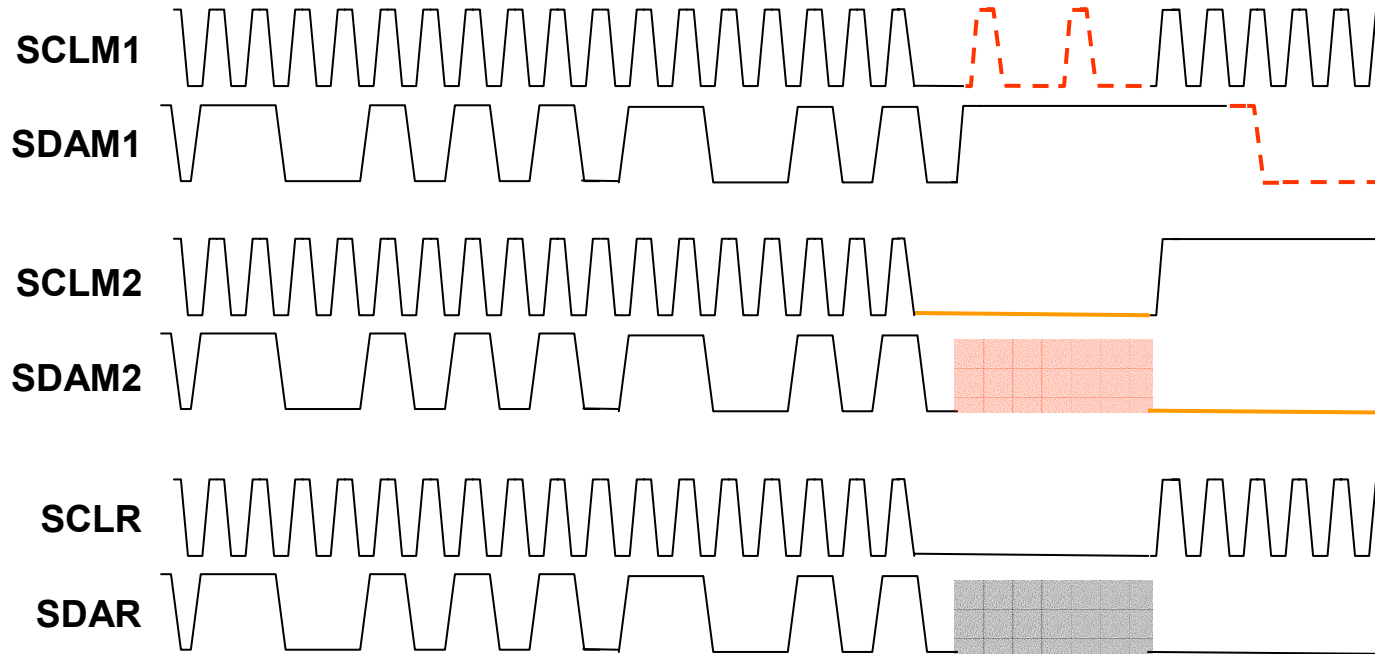


Problème : communication multi-maître

- M1 établit la communication avec M2. Pendant ACK, M2 sert une interruption et garde SCL au niveau bas. M1 attend SCL : time-out. M1 achève la transaction et réessaie d'établir la communication. Entre temps M2 redevient actif et relâche SCL, tout en gardant SDA bas : M1 ne peut pas redémarrer une transaction, car SDA est bas, et M2 ne peut pas achever la communication en cours, car SCL ne tombe pas



Synchronisation





I2C

Spécificité

Caractéristiques diverses du bus I2C



Les adresses réservées



Les adresses 0000 0xxx ne sont pas utilisées pour l'adressage des composants :



L'adresse 0000 0000 :

- Adresse d'appel général : les circuits ayant la capacité de traiter ce type d'appel émettent un acquittement. Le deuxième octet définit le contenu de l'appel
- 0000 0110 : RESET. Remet tous les registres des circuits connectés dans leur état initial. Les circuits qui le permettent rechargent leur adresse d'esclave.
- 0000 0010 : Les circuits qui le permettent rechargent leur adresse d'esclave.
- 0000 0100 : Les circuits définissant leur adresse de façon matérielle réinitialisent leur adresse d'esclave.
- 0000 0000 : interdit
- xxxx xxx1 : joue le rôle d'interruption. xxxx xxx peut être l'adresse du circuit qui a généré l'interruption



Les adresses réservées



Les adresses 0000 0xxx ne sont pas utilisées pour l'adressage des composants



Adresse 0000 0001 :

- Octet de start : utilisé pour synchroniser les périphériques lents avec les périphériques rapide



Adresse 0000 001x :

- Permet de rendre sourds tous les circuits I2C présents sur le bus : on peut donc changer le protocole de transmission sans générer d'erreurs au niveau des circuits I2C
- Le bus repasse en mode normal lors de la réception d'une condition d'arrêt



Adresses 0000 0110 à 0000 1111 :

- Non définies et ignorées par les circuits I2C : peuvent être utilisées pour debugger un réseau multimasters par exemple



I2C étendu



I2C Fast Mode :

- 400 kbits/s, Adressage sur 10 bits
- Paramètres physiques inchangés : protocole, niveaux, capacitance identiques : changement uniquement au niveau timing
- Abandon de la compatibilité CBUS
- Entrées à trigger de Schmitt
- Sorties haute impédance lorsque le périphérique n'est pas alimenté
- La résistance de tirage doit être adaptée
 - Jusqu'à 200pF : résistance suffit
 - De 200pF à 400pF : source de courant préférable

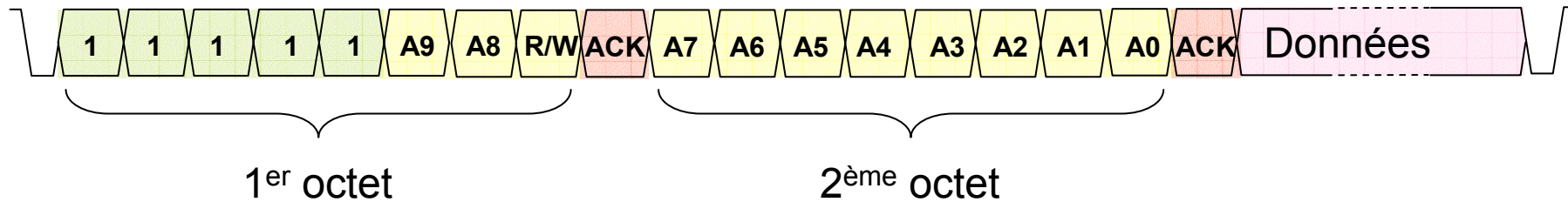


I2C étendu



Adressage étendu :

- Espace d'adressage trop restreint en mode standard
- 2 octets d'adressage :



- Compatibilité assurée avec le mode standard
 - R/W et ACK à la même position,
 - 11111 permet de faire la différence entre mode standard et mode fast