

4.1. Configuration minimaliste

Communication :

Microprocesseur



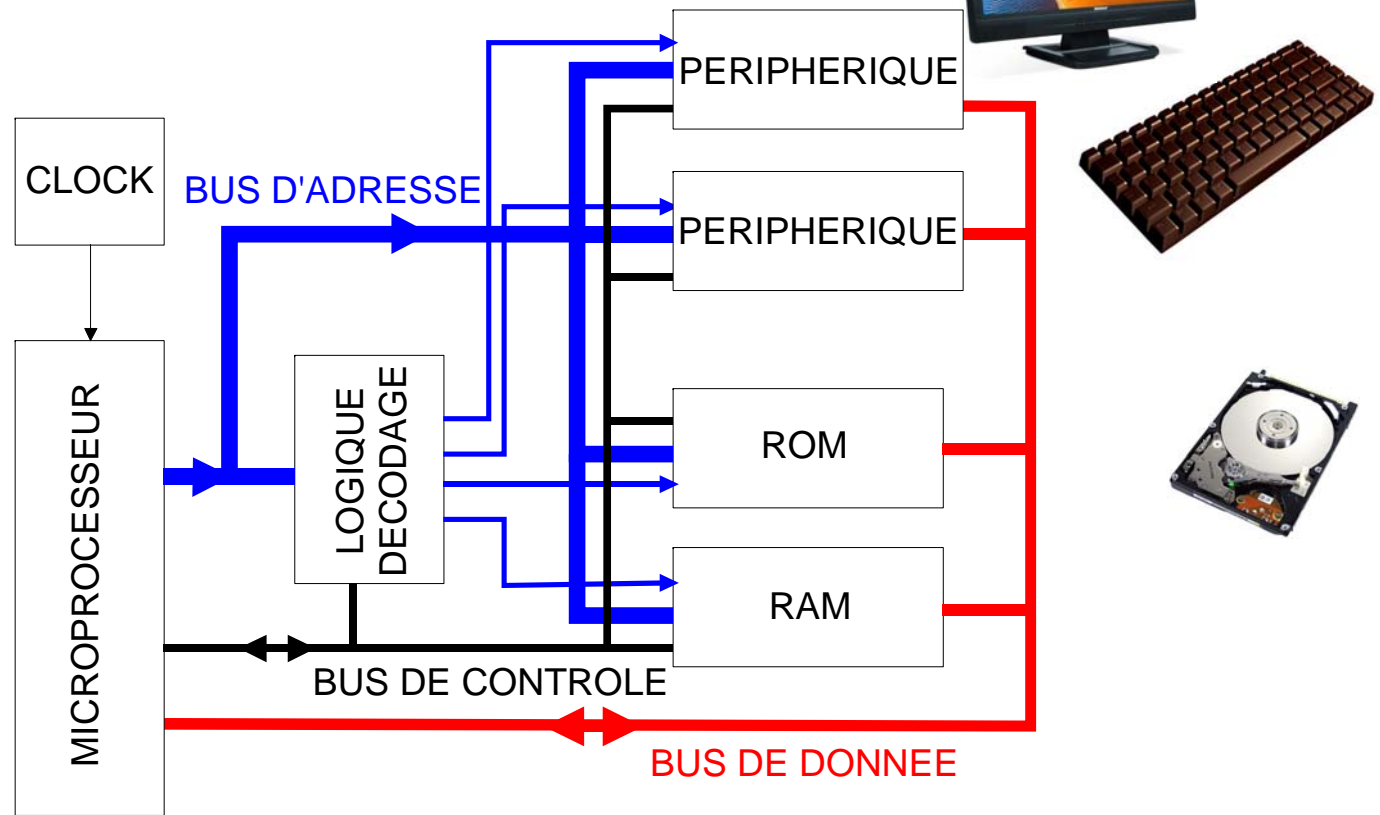
Mémoires

Périphériques

Microprocesseurs

Quatre groupes de signaux :

- Groupe 1 → signaux d'adresse
- Groupe 2 → signaux de donnée
- Groupe 3 → signaux de contrôle et d'état
- Groupe 4 → lignes d'alimentation et signal horloge



4.2. MOTOROLA 68000

Nombre de broches : 64

Signaux d'adresse :

A1-A23

Signaux de donnée :

D0-D15

Lignes d'alimentation et signal horloge :

V_{cc}, GND

CLK (4MHz, 6MHz, 8MHz)

Signaux de contrôle et d'état :

/AS, R/W

/UDS, /LDS, /DTACK

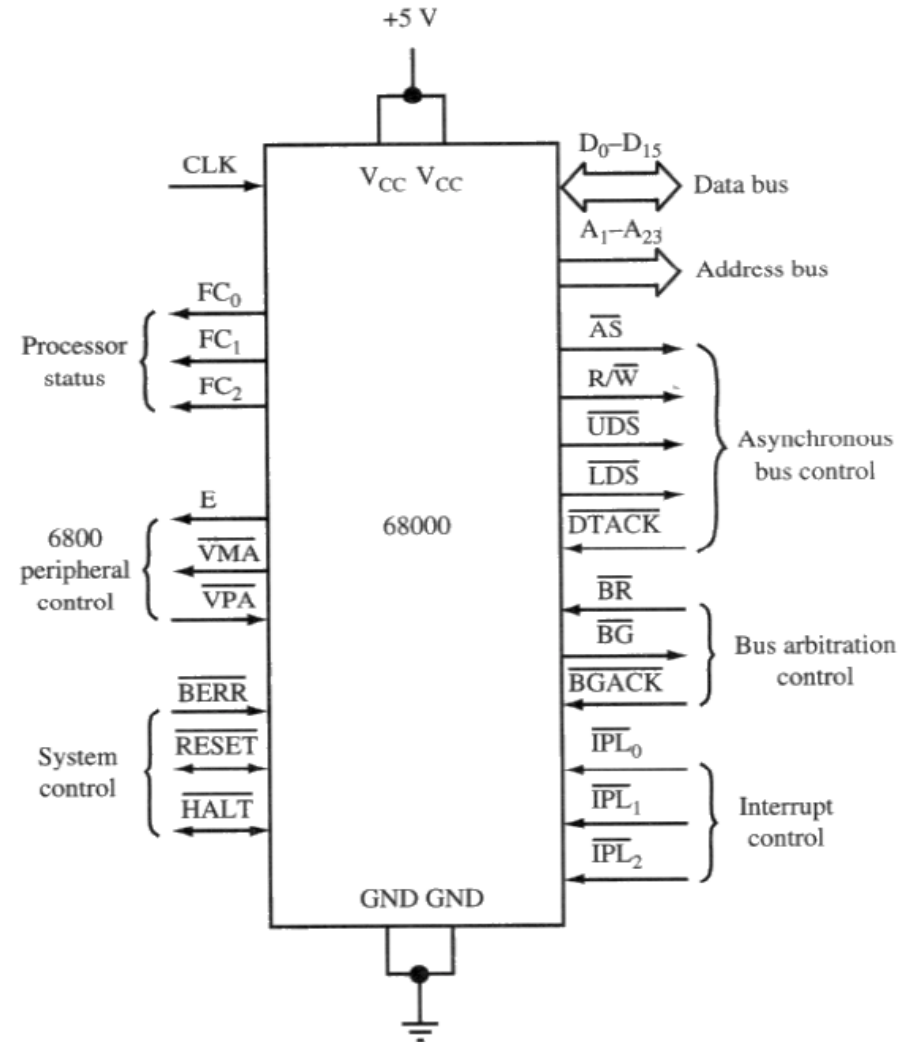
E, /VMA, /VPA

/BR, /BG, /BGACK

/IPL0, /IPL1, /IPL2

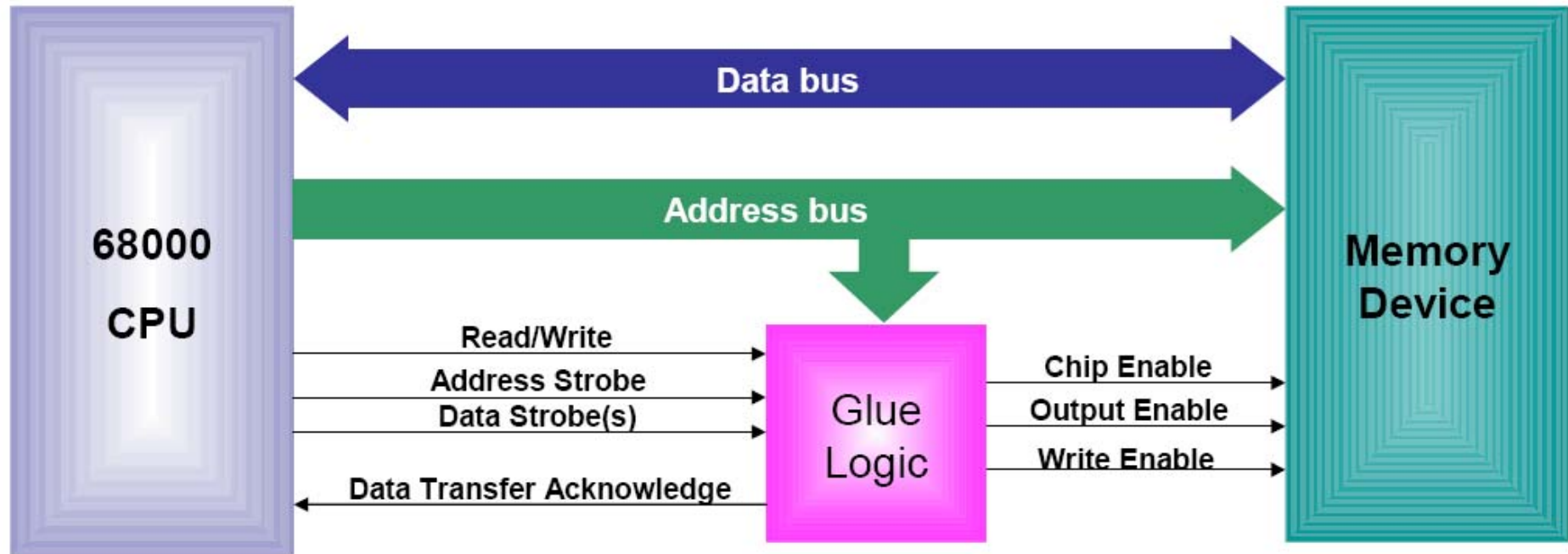
/FC0, /FC1, /FC2

/BERR, /RESET, /HALT



Ceramic DIP

4.3. Echanges de données avec les circuits mémoire ou périphérique



Bus d'adresse (24 bits)

- (A0)-A1-A23 (A0 signal interne)
- Spécifie l'emplacement de la lecture ou de l'écriture
- Espace addressable : $2^{24} = 16$ Moctets
- Le signal A0 est interne et utilisé pour piloter les signaux /UDS et /LDS
- Sortie à trois états → un autre circuit peut prendre le contrôle du bus
- Fonctionne différemment durant la prise en charge d'une interruption

Bus de donnée (16 bits)

- D0-D15
- Pour le transfert des données vers les circuits mémoire et périphérique ou en provenance de ceux-ci
- Lignes bidirectionnelles. Elles sont positionnées :
 - En entrée durant un cycle CPU de lecture
 - En sortie durant un cycle CPU d'écriture
- La totalité des lignes est utilisée lors d'une opération sur un mot
- La moitié des lignes est utilisée lors d'une opération sur un octet (D0-D7 ou D8-D15)
- Durant une interruption vectorisée, le vecteur d'interruption est placé sur les lignes D0-D07

Format de transfert des données

Transfert en format **mot** (16 bits) ou **octet** (8 bits)

Espace d'adressage du 68000

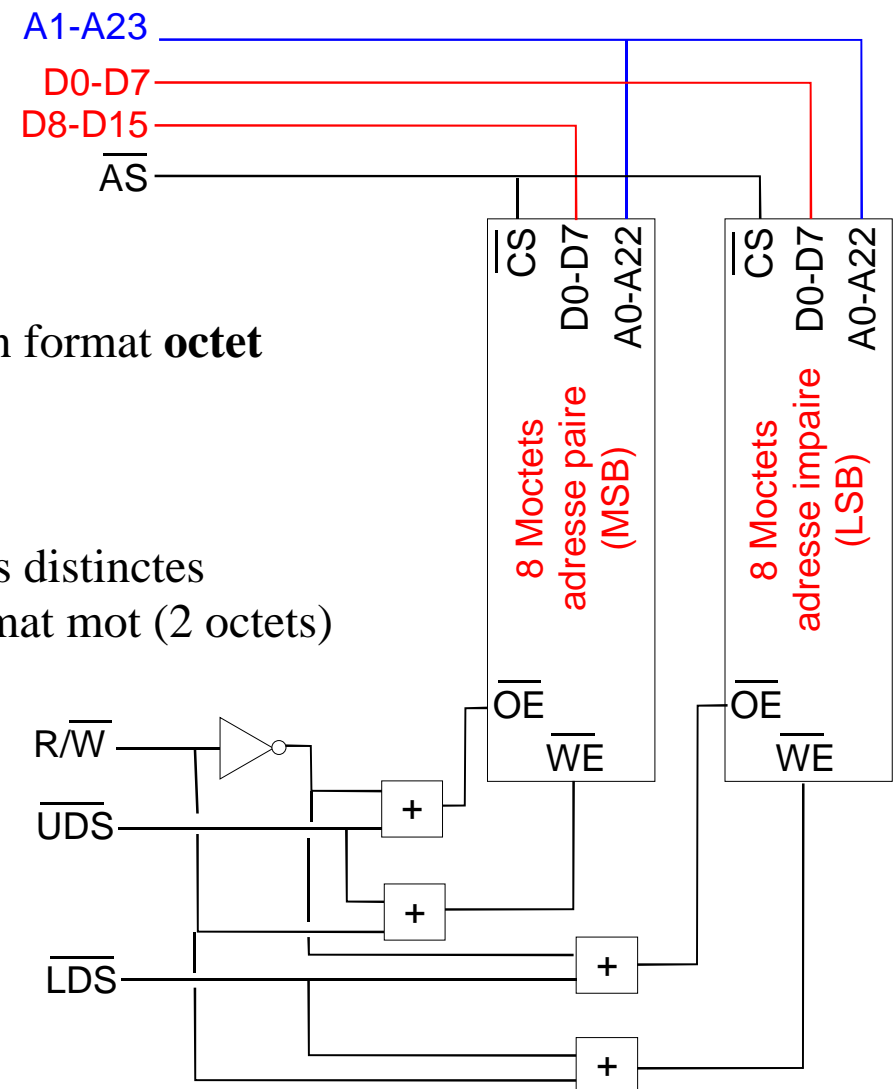
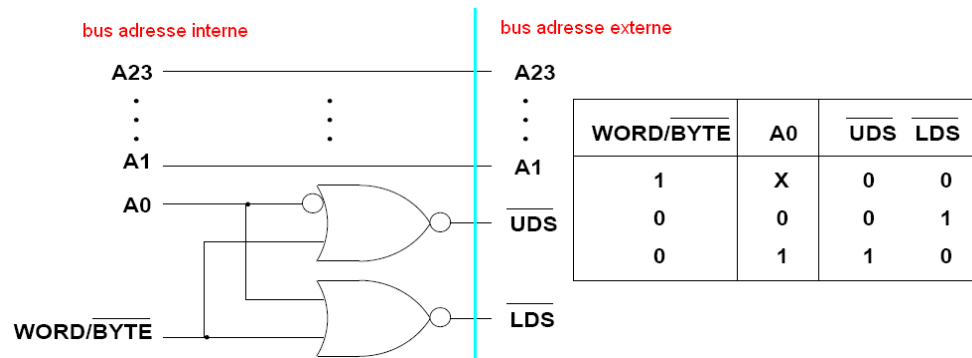
→ $2^{24} = \mathbf{16\ 777\ 216}$ adresses distinctes / transfert en format **octet**

Bus de donnée : 16 bits

→ $2^{23} = 8\ 388\ 608$ adresses distinctes

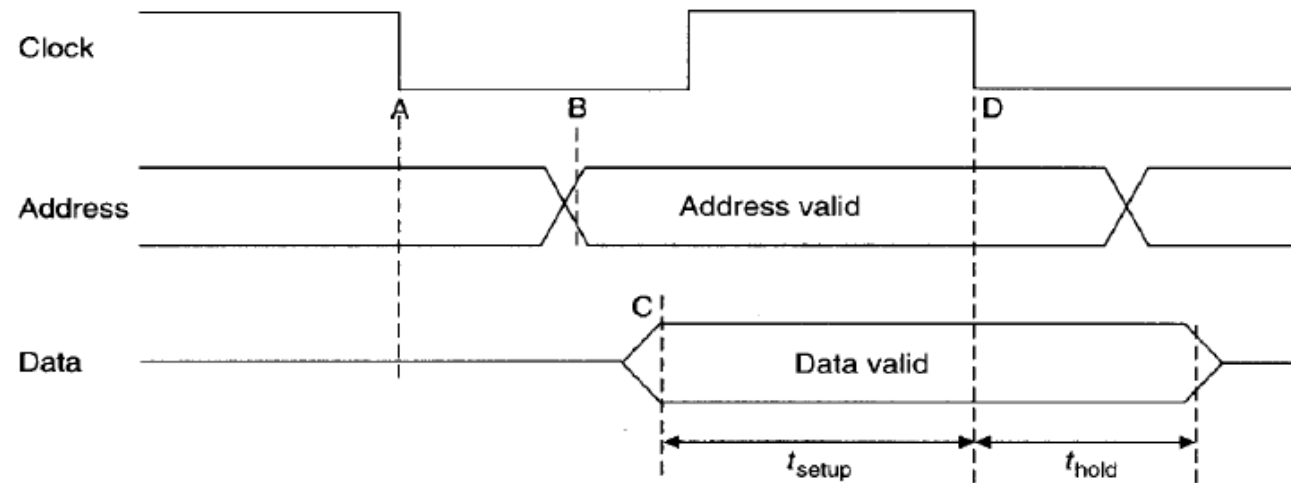
Bus d'adresse : 23 bits

/ transfert en format mot (2 octets)

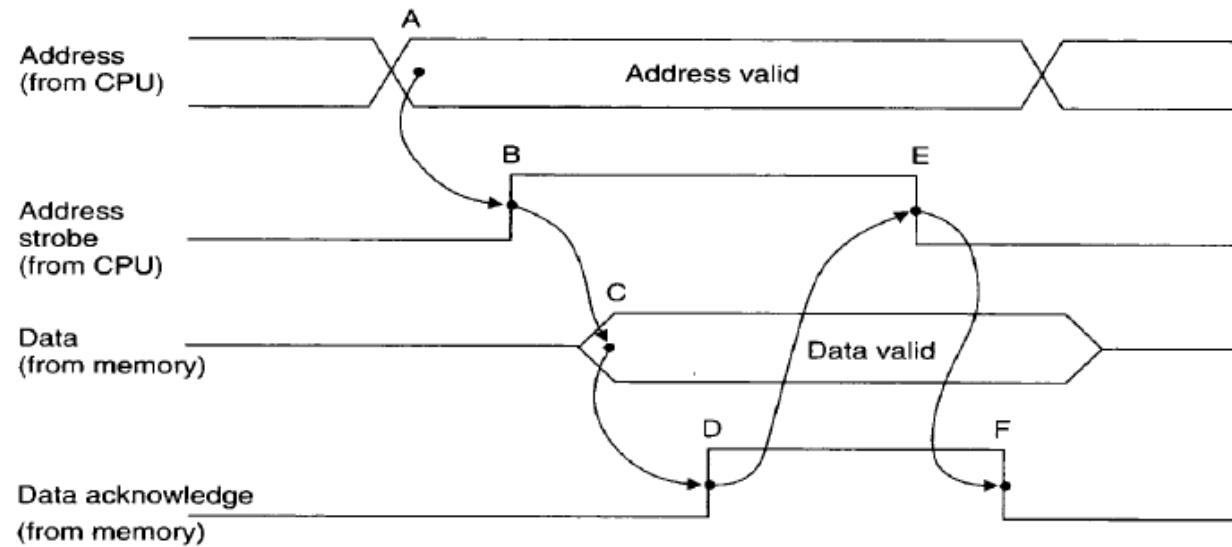


Echanges synchrone et asynchrone

Synchronous



Asynchronous (handshake)



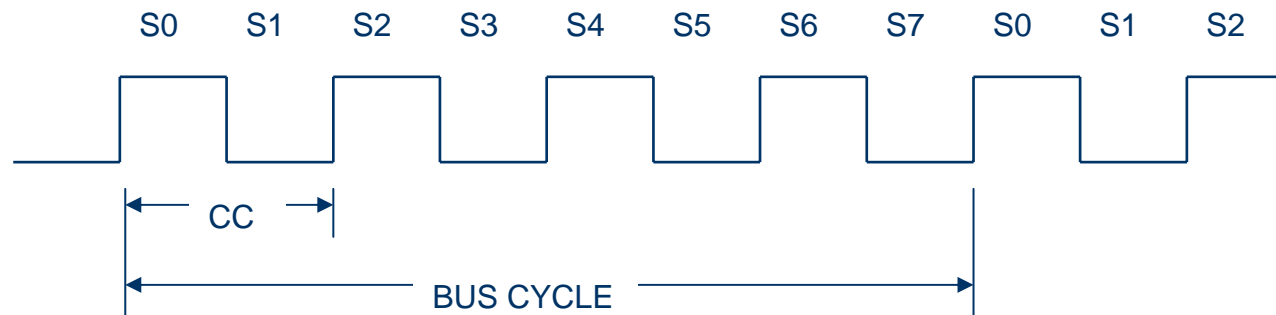
Echange asynchrone

Lorsque qu'un cycle de bus est initié par le CPU pour une écriture ou une lecture, il ne sera achevé que lorsque ce dernier aura reçu une réponse de la mémoire ou du périphérique. Cette réponse est un signal d'acquiescement qui indique au CPU que le cycle de bus actuel est complété.

La séquence des opérations est appelée poignée de mains (handshake)

Définitions

Cycle horloge (cc)	→ 1 période du signal horloge
Etat du bus (Sn)	→ $\frac{1}{2}$ période du signal horloge
Cycle de bus	→ temps pour accomplir une lecture ou écriture
Cycle d'instruction	→ temps pour lire, décoder et exécuter une instruction



Signaux mis en œuvre

/AS (Address Strobe)

- sortie
- affirmé pour indiquer qu'une adresse valide est placée sur le bus d'adresse

/UDS (Upper Data Strobe) and **/LDS** (Lower Data Strobe)

- sortie à 3 états
- détermine le format (taille) de la donnée échangée :
 - en format mot → $\text{/UDS} = \text{/LDS} = 0$ (adresse paire)
 - en format octet → $\text{/LDS} = 0$ et $\text{/UDS} = 1$ (adresse impaire – lignes D0-D7 utilisées)
 $\text{/LDS} = 1$ et $\text{/UDS} = 0$ (adresse paire – lignes D80-D15 utilisées)

/DTACK (Data Transfer ACKnowledge)

- entrée (le seul signal de contrôle en entrée)
- signal de poignée de mains (handshake) généré par le circuit adressé
- wait states
- le CPU attend jusqu'à ce que /DTACK soit affirmé :
 - cycle de lecture : la donnée sur le bus est mémorisé lorsque /DTACK est affirmé
 - cycle d'écriture : la donnée est maintenue sur le bus tant que /DTACK n'est pas affirmé

R/W (Read/Write)

- sortie à 3 états
- spécifie la nature du cycle de bus
- $R/W = 1 \rightarrow$ lecture
- $R/W = 0 \rightarrow$ écriture
- pour éviter une écriture intempestive, le CPU place R/W à 1 lorsqu'il réalise une opération interne
- placé à l'état haute impédance lorsque le CPU rend la main au niveau du bus mémoire

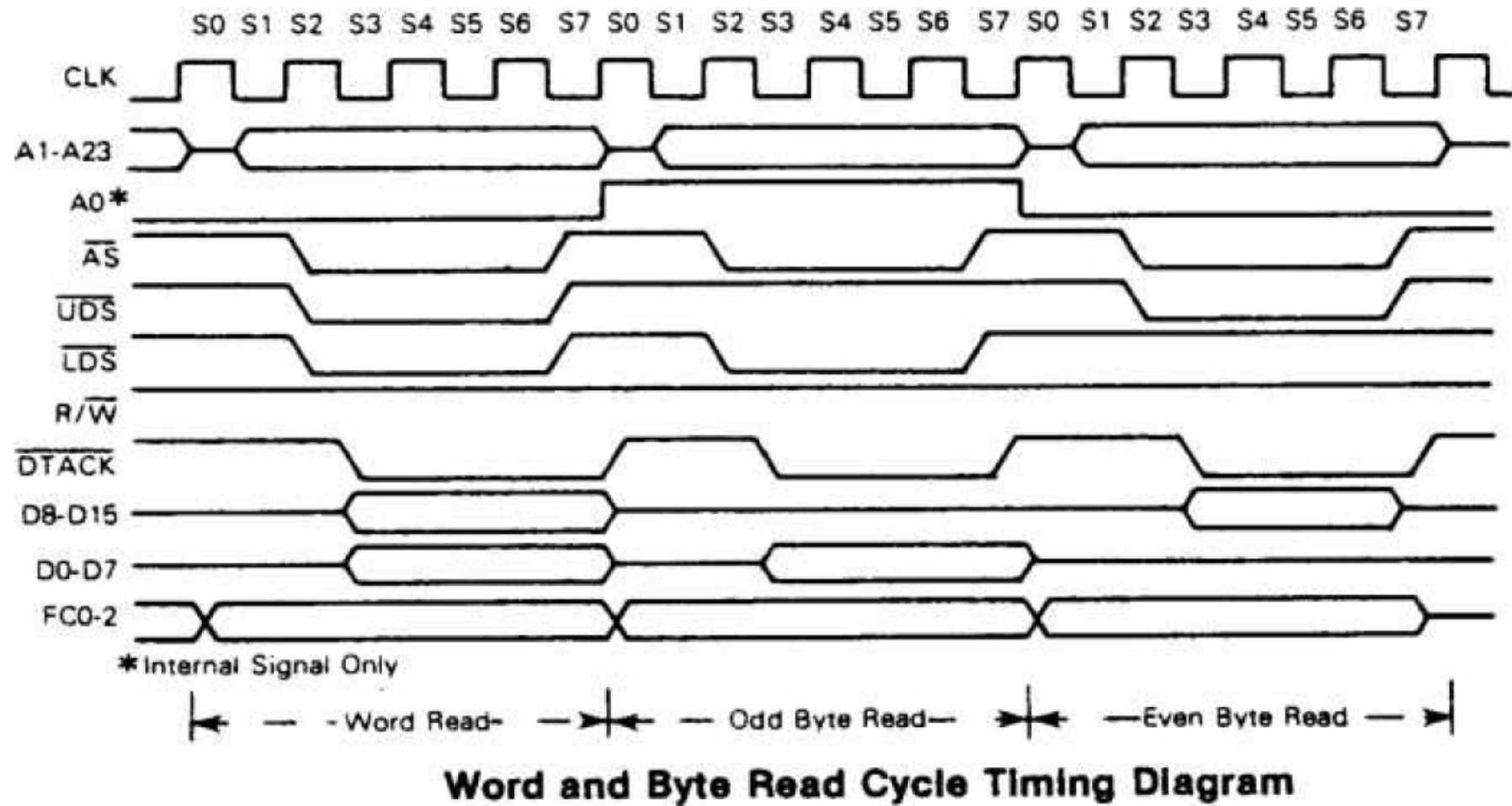
Principe du transfert asynchrone

- le CPU place l'adresse sur le bus et affirme /AS pour signaler aux mémoires et périphériques qu'une adresse valide est disponible sur le bus
- la mémoire ou le périphérique (adressé) affirme /DTACK pour informer le CPU :
 - lors d'une opération de lecture qu'une donnée valide est positionnée sur le bus
 - lors d'une opération d'écriture que la donnée a été écrite avec succès dans la mémoire ou le périphérique

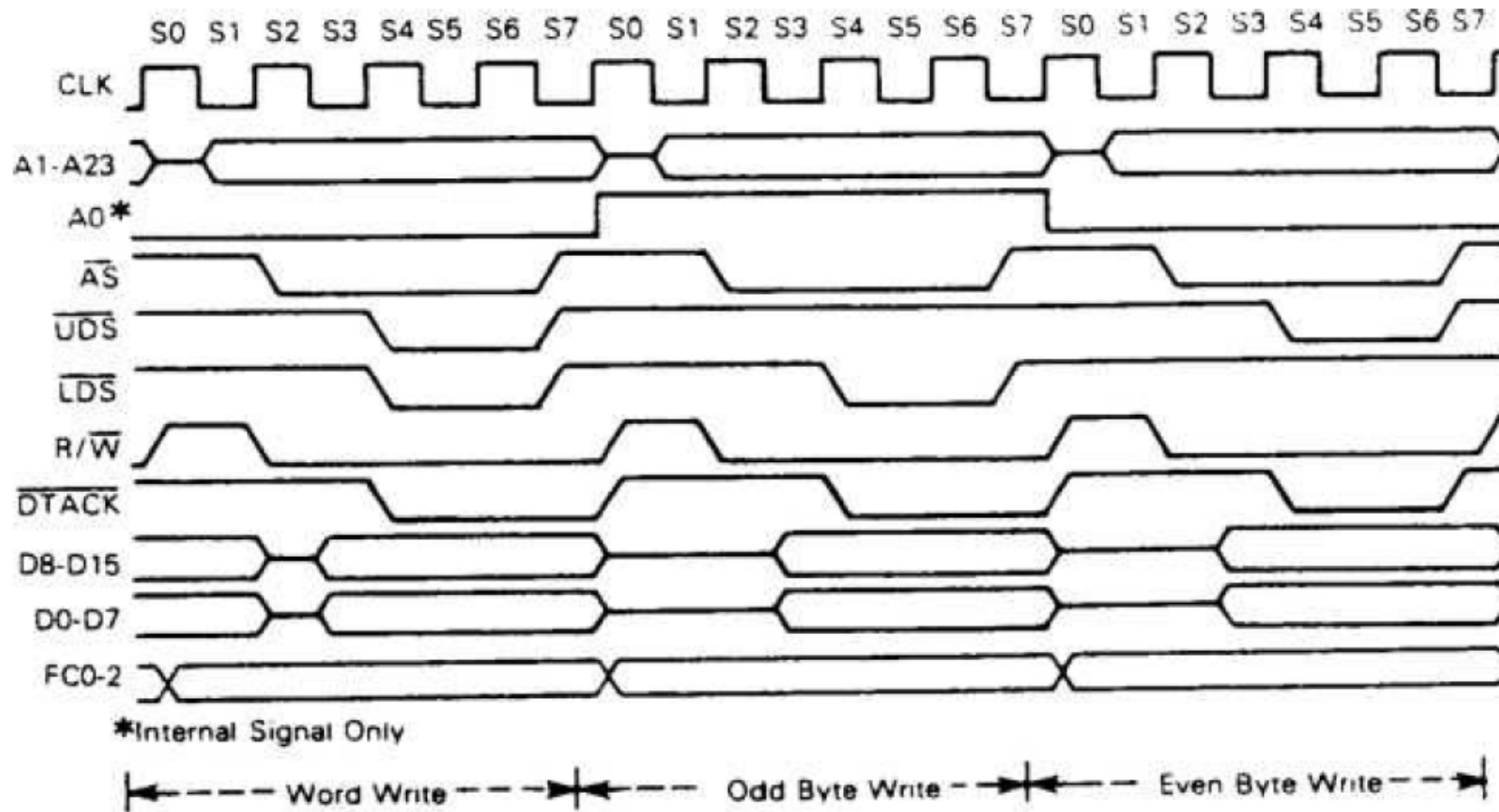
Un accès mémoire occupe au minimum 8 états horloge (S0 à S7)

L'opération asynchrone fonctionne également avec des circuits (lents) ayant avec un long temps d'accès. Dans ce cas des états d'attente (**wait state**) sont insérés dans le cycle de bus

Chronogrammes des accès en lecture

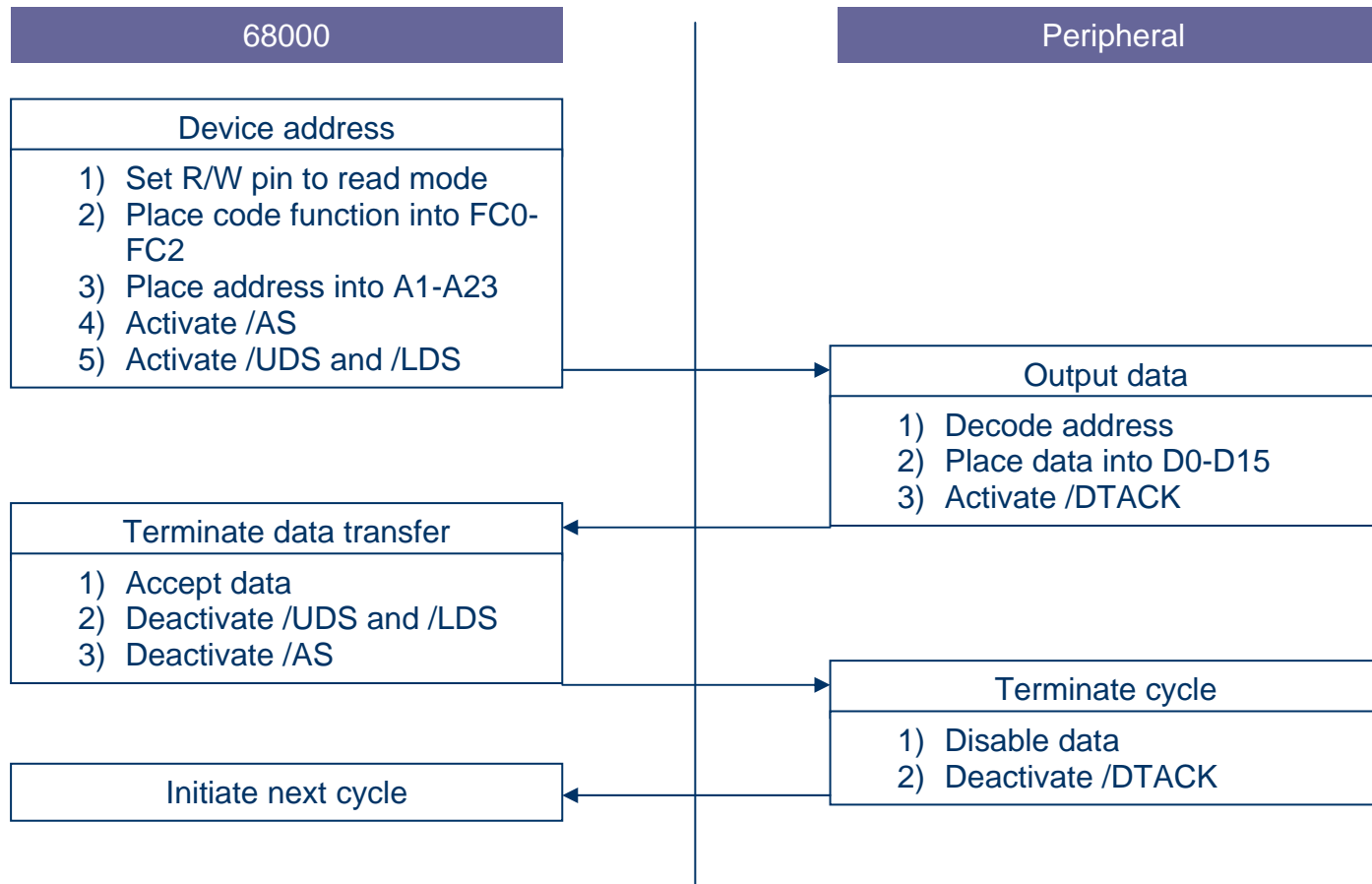


Chronogrammes des accès en écriture



Word and Byte Write Cycle Timing Diagram

Séquencement des opérations pour un accès en lecture



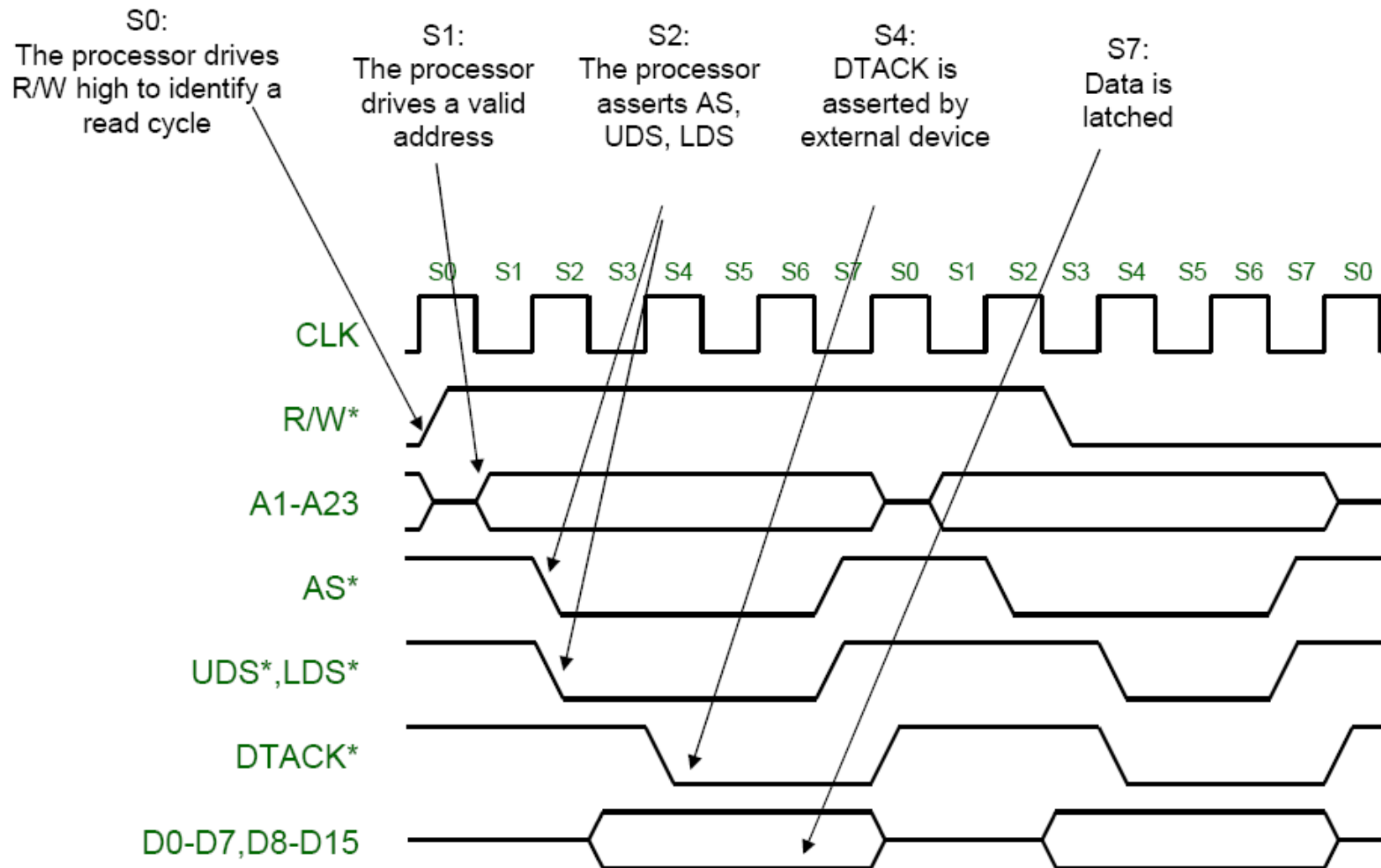
Décomposition dans le temps de la séquence

- STATE 0 The read cycle starts in state 0 (S0). The processor places valid function codes on FC0–FC2 and drives R/W high to identify a read cycle.
- STATE 1 Entering state 1 (S1), the processor drives a valid address on the address bus.
- STATE 2 On the rising edge of state 2 (S2), the processor asserts /AS and /LDS, and/or /UDS.
- STATE 3 During state 3 (S3), no bus signals are altered.
- STATE 4 During state 4 (S4), the processor waits for a cycle termination signal (/DTACK or /BERR) or /VPA, an M6800 peripheral signal. When /VPA is asserted during S4, the cycle becomes a peripheral cycle. If neither termination signal is asserted before the falling edge at the end of S4, the processor inserts wait states (full clock cycles) until either /DTACK or /BERR is asserted.

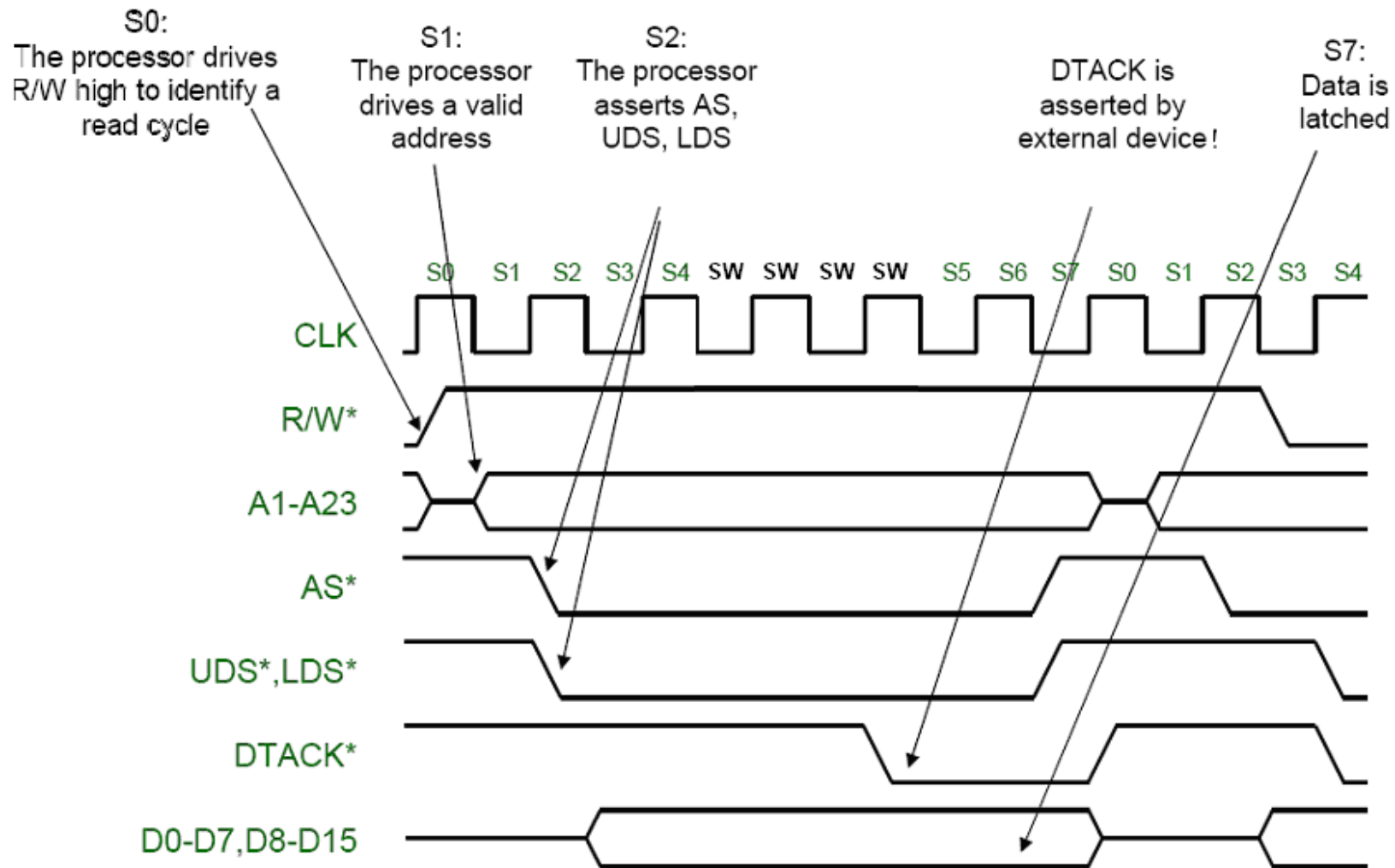
- STATE 5 During state 5 (S5), no bus signals are altered.
- STATE 6 During state 6 (S6), data from the device is driven onto the data bus.
- STATE 7 On the falling edge of the clock entering state 7 (S7), the processor latches data from the addressed device and negates /AS and /LDS, and /UDS. At the rising edge of S7, the processor places the address bus in the high impedance state. The device negates /DTACK or /BERR at this time.

NOTE : During an active bus cycle, VPA and BERR are sampled on every falling edge of the clock beginning with S4, and data is latched on the falling edge of S6 during a read cycle. The bus cycle terminates in S7, except when BERR is asserted in the absence of DTACK. In that case, the bus cycle terminates one clock cycle later in S9.

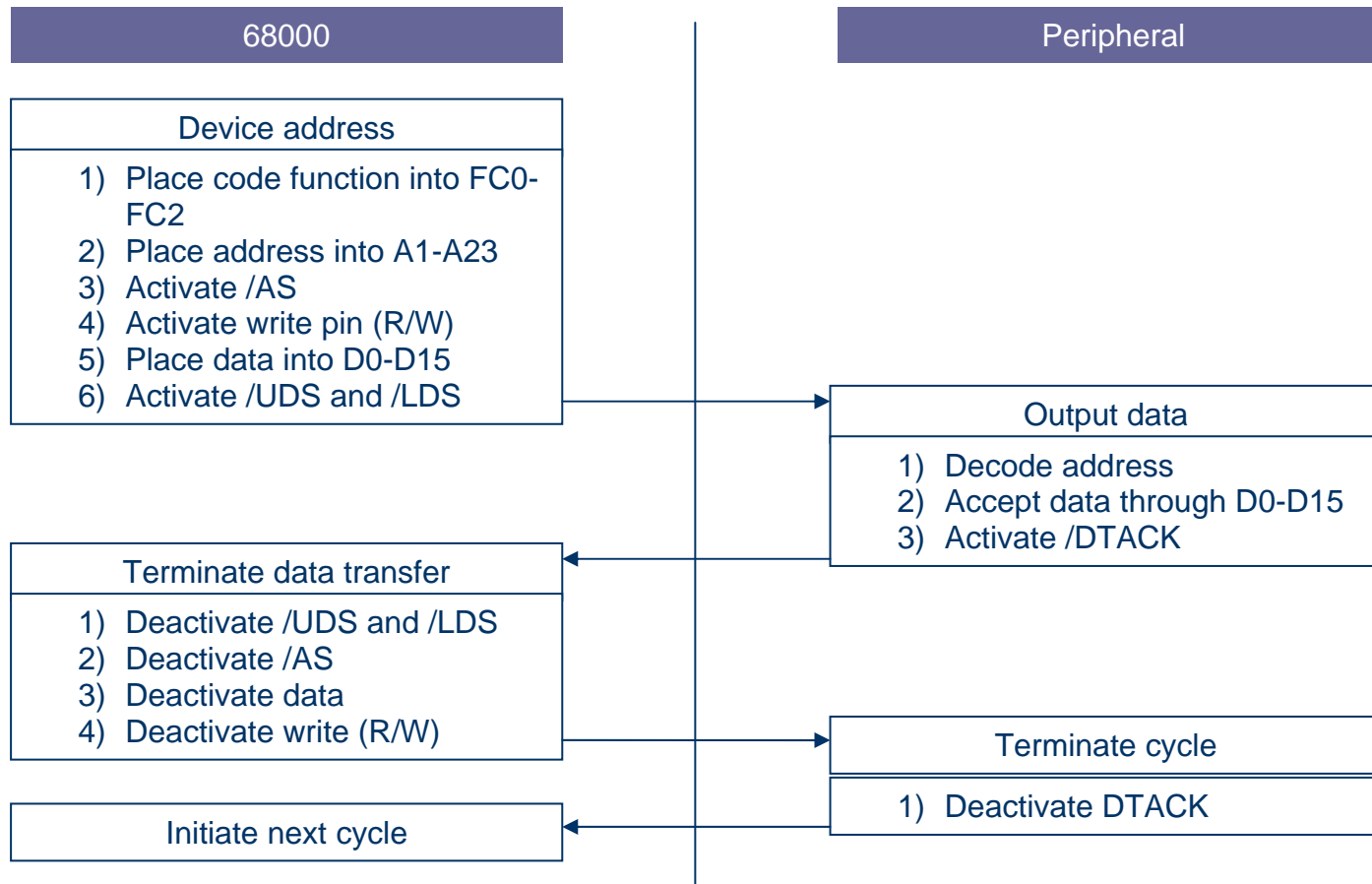
Accès en lecture - Commentaires des chronogrammes



Accès en lecture – Insertion d'états d'attente (wait state)



Séquencement des opérations pour un accès en écriture



Décomposition dans le temps de la séquence

- STATE 0 The write cycle starts in S0. The processor places valid function codes on FC2–FC0 and drives R/W high (if a preceding write cycle has left R/W low).
- STATE 1 Entering S1, the processor drives a valid address on the address bus.
- STATE 2 On the rising edge of S2, the processor asserts /AS and drives R/W low.
- STATE 3 During S3, the data bus is driven out of the high-impedance state as the data to be written is placed on the bus.
- STATE 4 At the rising edge of S4, the processor asserts /LDS, and/or UDS. The processor waits for a cycle termination signal (/DTACK or /BERR) or /VPA, an M6800 peripheral signal. When /VPA is asserted during S4, the cycle becomes a peripheral cycle. If neither termination signal is asserted before the falling edge at the end of S4, the processor inserts wait states (full clock cycles) until either /DTACK or /BERR is asserted.

4. MICROPROCESSEUR 68000 (MATERIEL) – SIGNAUX D'ECHANGES

STATE 5 During S5, no bus signals are altered.

STATE 6 During S6, no bus signals are altered.

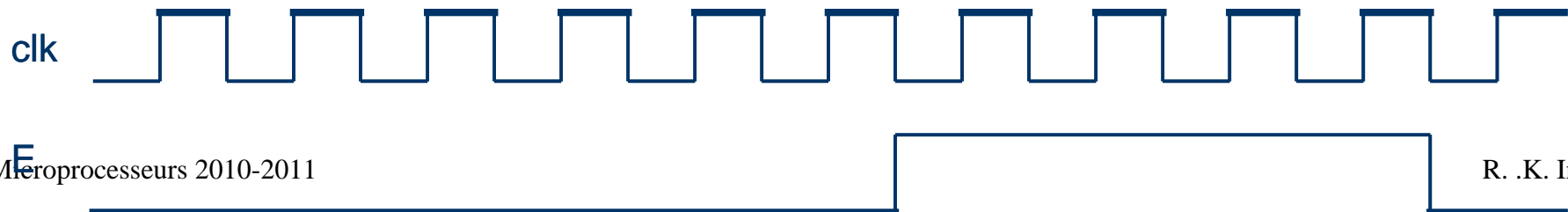
STATE 7 On the falling edge of the clock entering S7, the processor negates /AS, /LDS, and /UDS. As the clock rises at the end of S7, the processor places the address and data buses in the high-impedance state, and drives R/W high. The device negates /DTACK or /BERR at this time.

Echange synchrone

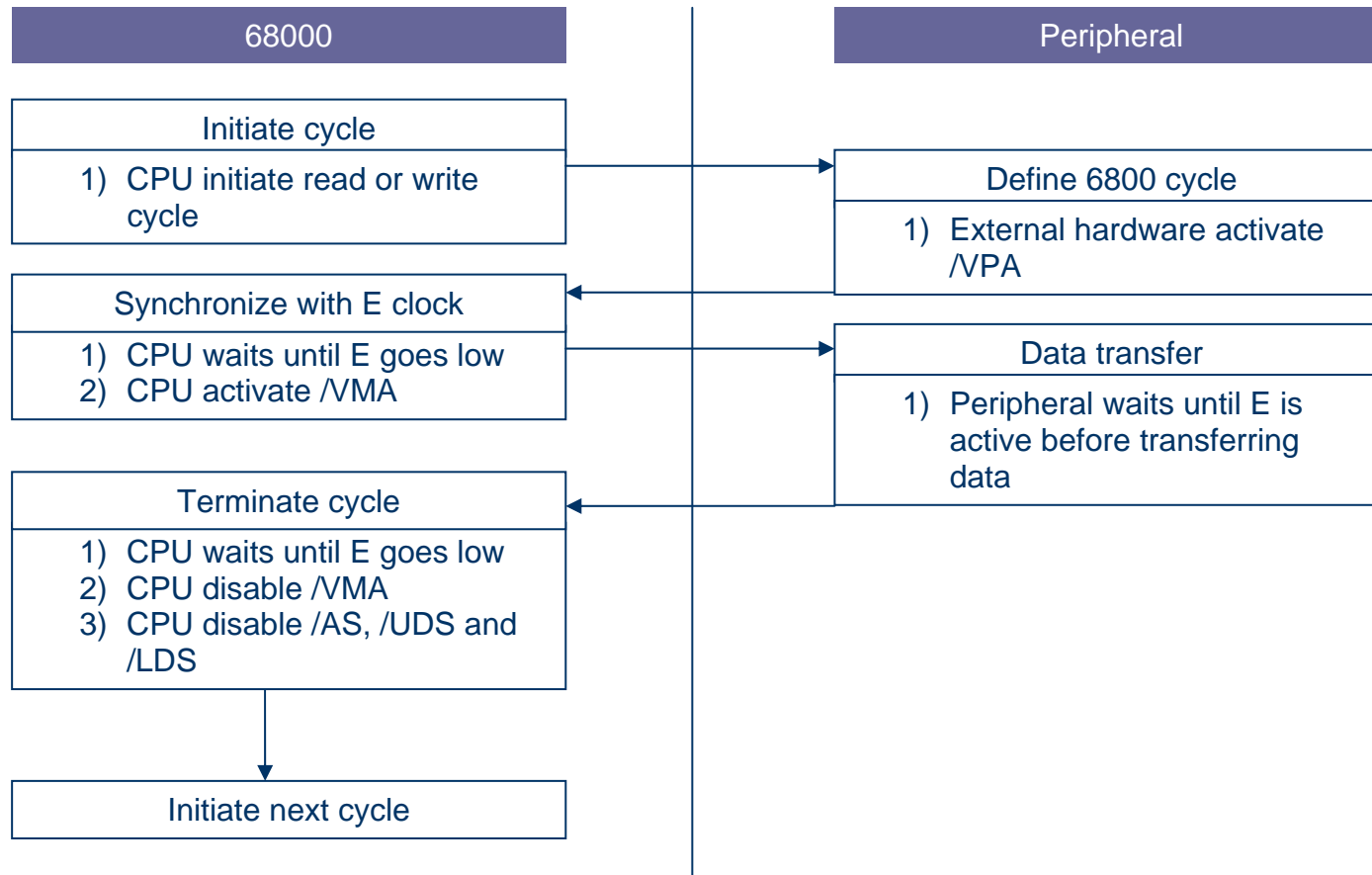
Permet au 68000 d'utiliser des circuits de la précédente génération 6800 tels que le PIA 6821 ou l'ACIA 6850.

Signaux mis en œuvre

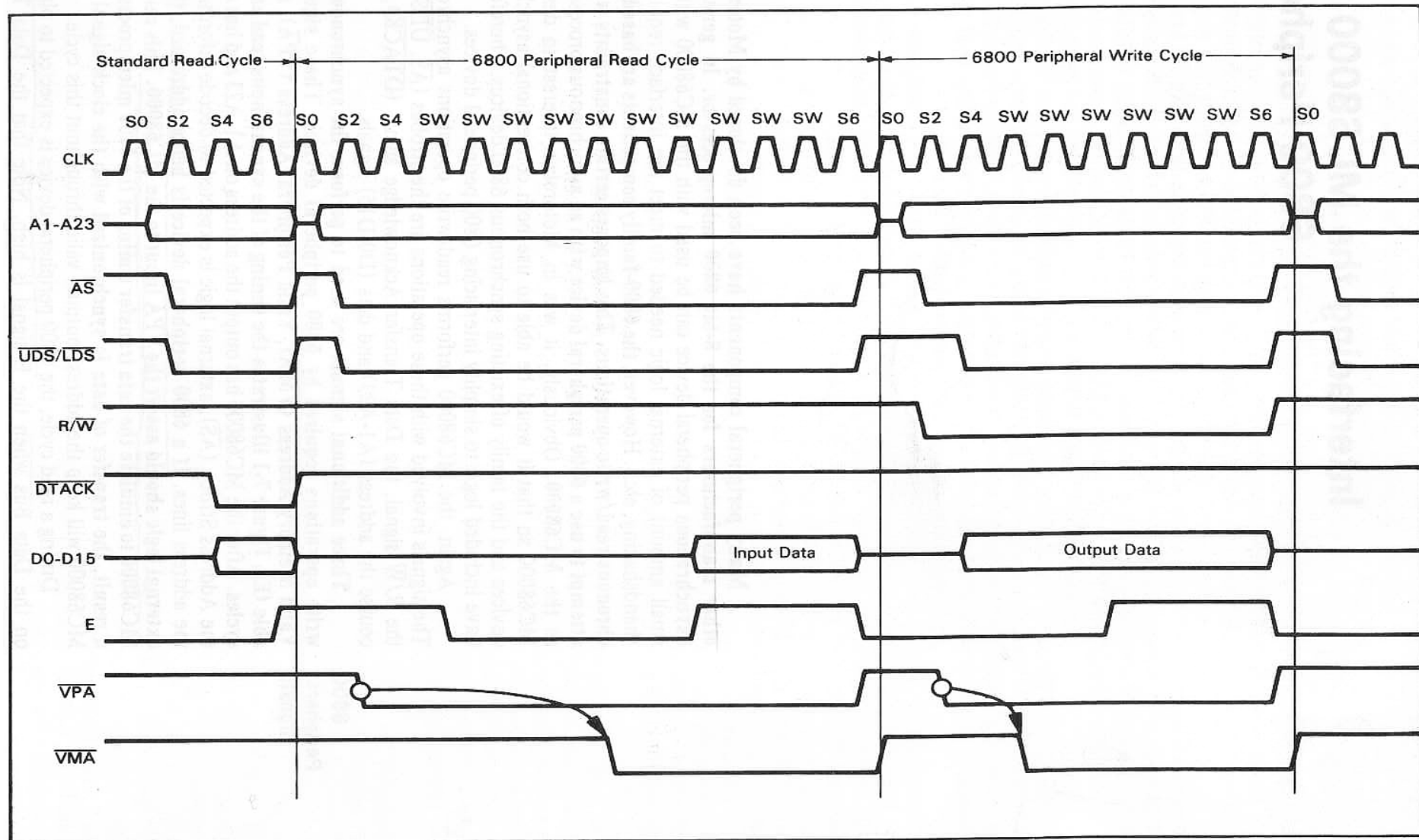
- E (enable clock)
 - sortie
 - signal indépendant de l'état du bus du CPU
 - signal de synchronisation utilisé par les périphériques de génération 6800
 - horloge CPU divisée par 10 → 6 cycles à l'état bas et 4 cycles à l'état haut
- /VPA (valid peripheral address)
 - entrée
 - informe le 68000 qu'un périphérique de génération 6800 est en train d'être adressé
- /VMA (valid memory address)
 - sortie
 - informe le périphérique concerné qu'une adresse valide est placée sur le bus et que le CPU est synchronisé sur E
 - en réponse au signal /VPA



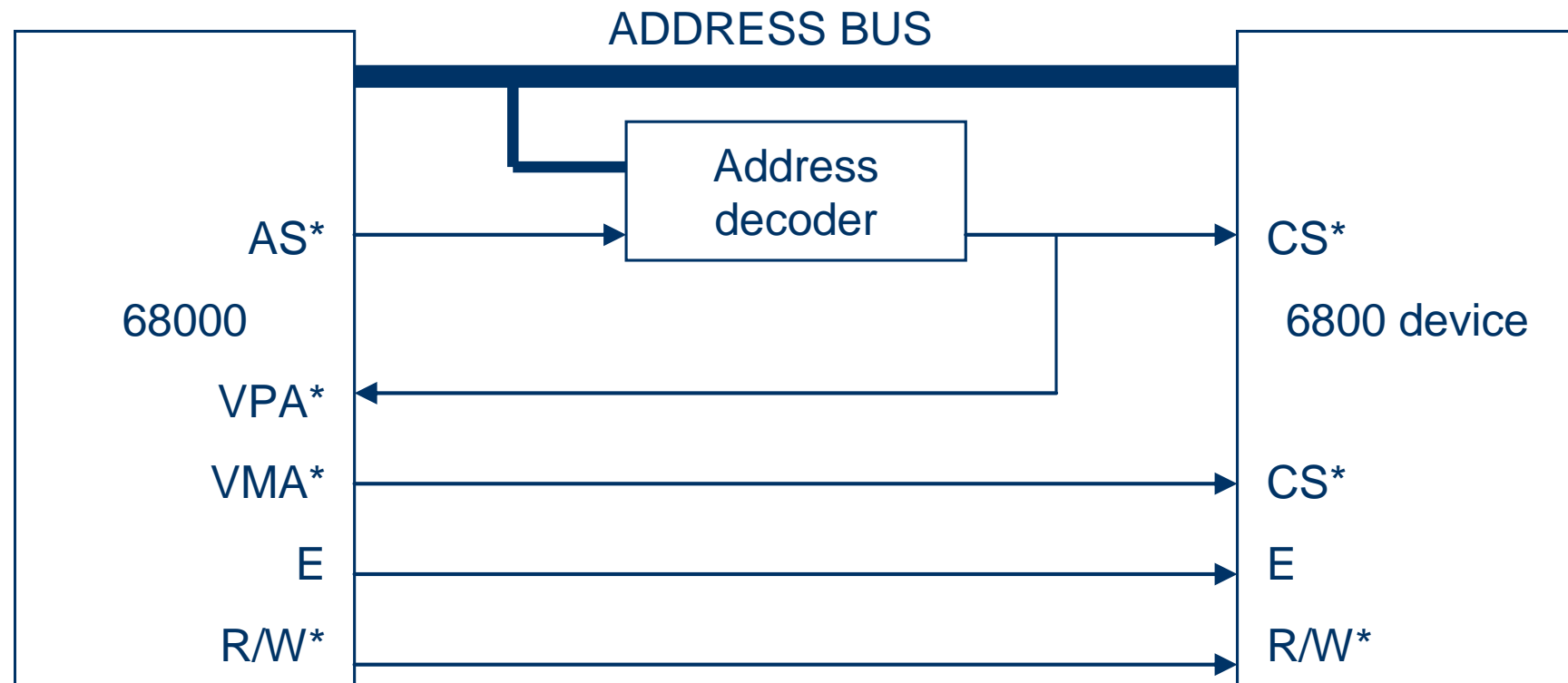
Séquencement des opérations pour un accès en lecture ou écriture



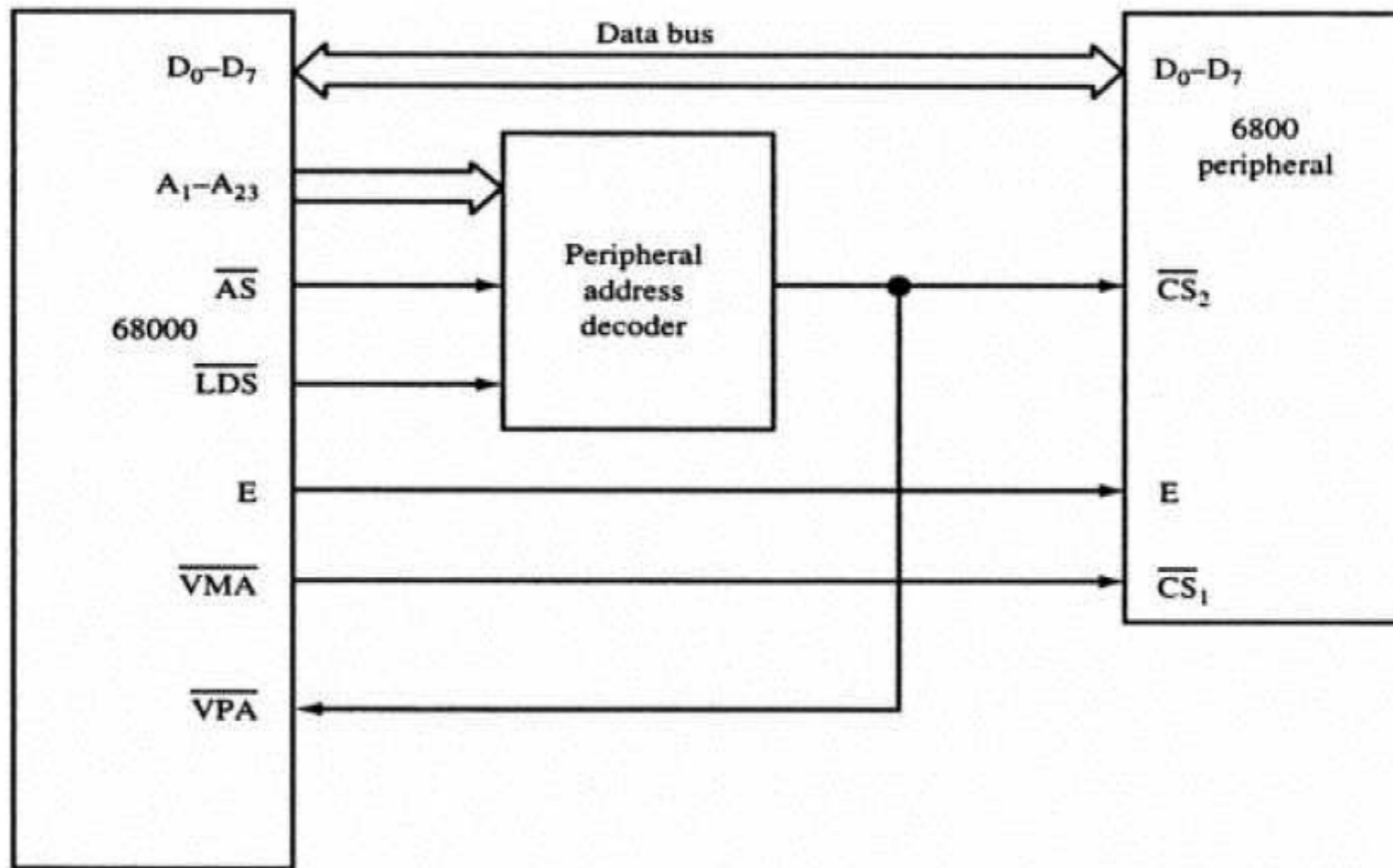
Chronogrammes des accès en lecture et écriture



Exemple 1 d'interfaçage avec un périphérique de génération 6800

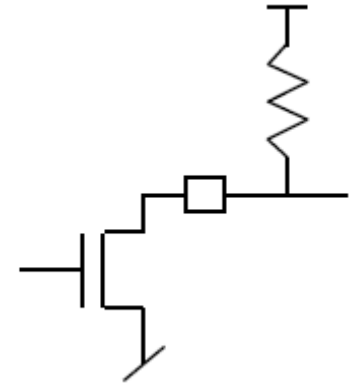


Exemple 2 d'interfaçage avec un périphérique de génération 6800

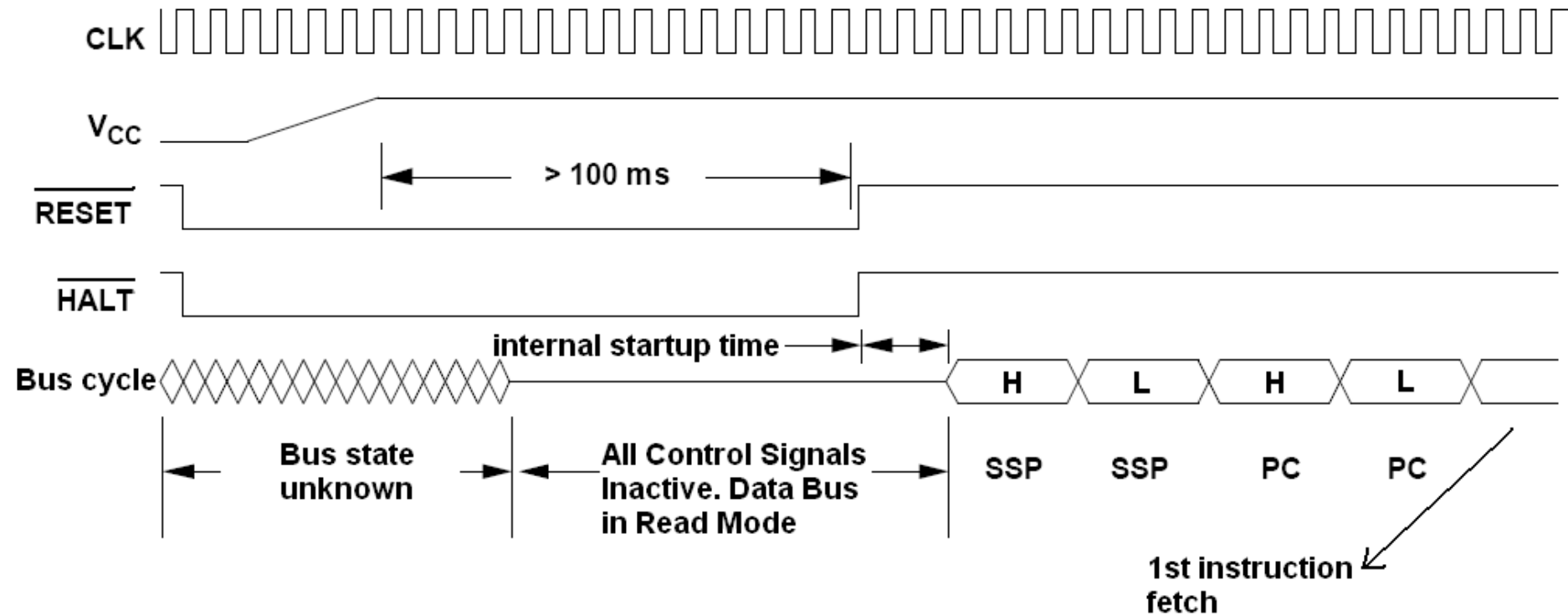


4.4. Signaux de contrôle du système /RESET

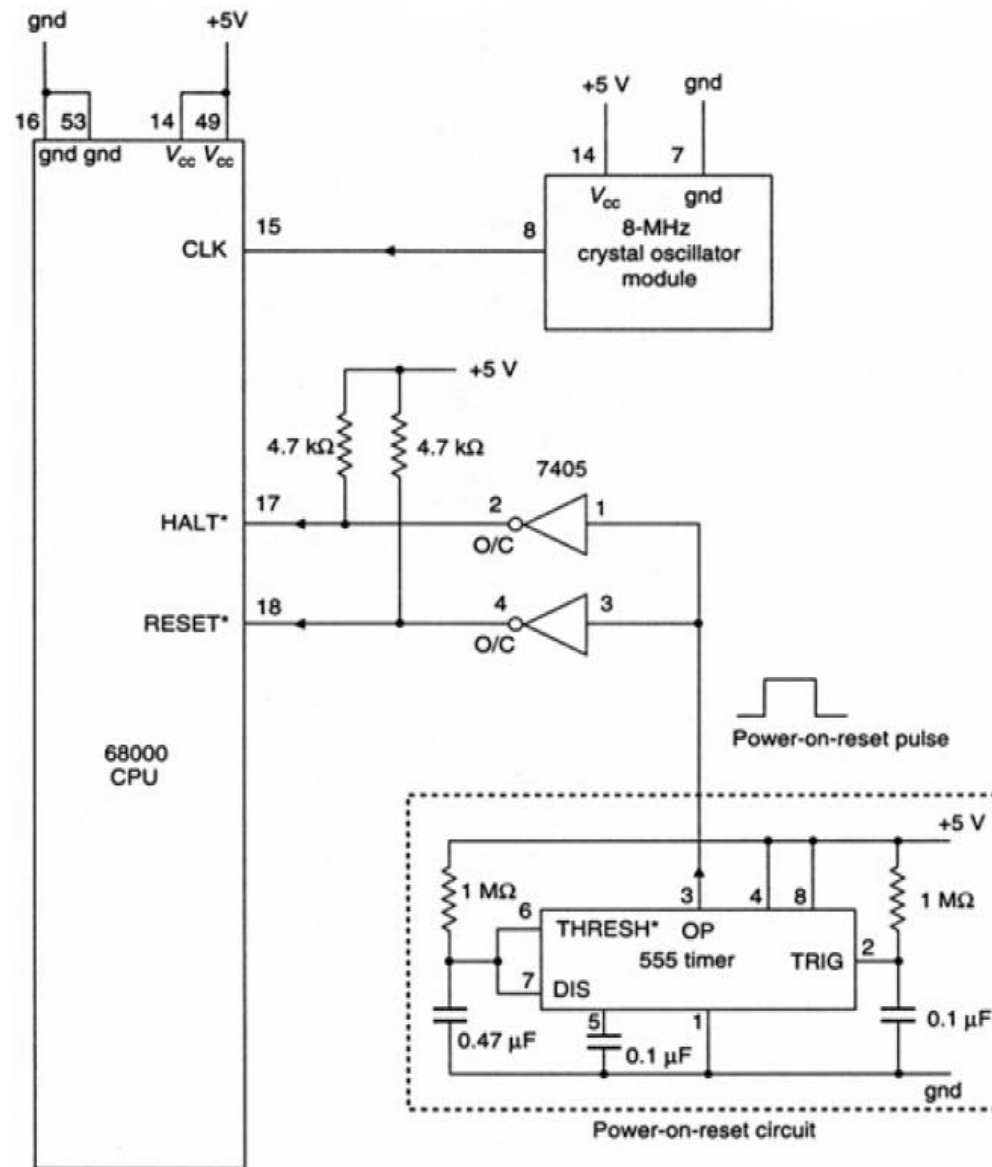
- signal bi-directionnel utilisé pour amener le CPU dans un état déterminé
- entrée/sortie active à l'état bas
 - drain ouvert – résistance externe de type “pull up” à placer
 - peut former un OU câblé avec d'autres entrées
- utilisé en entrée :
 - charge SSP et PC en format mot long avec les contenus mémoire d'adresse \$000000 et \$000004
 - au moment de la mise sous tension, /RESET et /HALT doivent être maintenus à l'état bas pendant au minimum 100 ms.
 - autrement, /RESET and /HALT doivent être affirmés pendant un minimum de 10 cycles horloge. L'entrée /RESET est en général relié à un bouton poussoir de réinitialisation du CPU.
- utilisé en sortie :
 - le 68000 dispose de l'instruction RESET
 - lorsque l'instruction est exécutée, le CPU affirme la sortie /RESET pendant 124 cycles horloge → autorise le CPU à réinitialiser son environnement sans se réinitialiser lui-même.
- dans les systèmes basiques, /HALT et /RESET sont reliés.



Chronogrammes de mise sous tension



Electronique de RESET



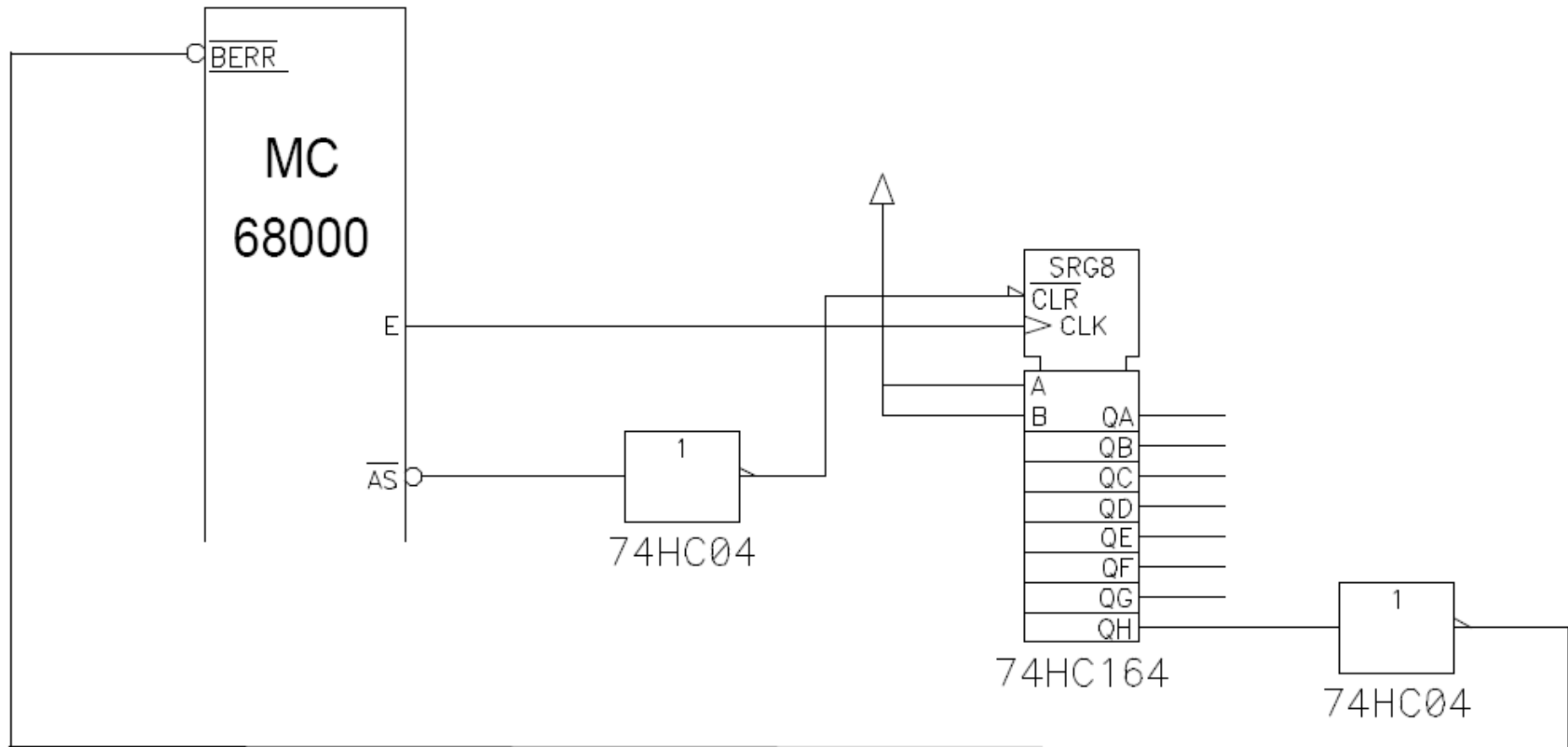
/HALT

- signal bi-directionnel / entrée sortie drain ouvert
- lorsqu'il est affirmé par un circuit externe
 - le CPU stoppe à la fin du cycle de bus courant
 - le CPU se réexécute à nouveau lorsque /HALT repasse à l'état haut.
- /HALT peut être utilisé pour forcer le 68000 à n'exécuter qu'un seul cycle de bus à la fois
 - mode d'usage appelé "pas à pas"
 - utilisé pour déverminer (debugger) un système
 - /HALT doit être affirmé juste assez longtemps pour permettre le CPU de n'exécuter qu'un seul cycle de bus.
- /HALT peut être utilisé pour répéter un cycle de bus en cas d'erreur d'accès mémoire
 - /HALT peut être affirmé soit avant soit pendant l'affirmation de /BERR
 - lorsque /HALT et /BERR sont affirmés, le CPU termine le cycle de bus courant
 - lorsque /HALT est désactivé, le CPU exécute le précédent cycle
- /HALT en sortie :
 - lorsque le 68000 se trouve dans une situation dans laquelle il ne peut se récupérer (double erreur de bus)
 - le CPU stoppe et affirme le signal /HALT pour indiquer le problème
 - l'instruction STOP peut également démarrer la condition de halte et positionner la sortie /HALT à l'état bas.

/BERR

- le système utilise /BERR pour informer le 68000 un dysfonctionnement (exemple : le CPU tente d'accéder à une adresse mémoire invalide)
- le 68000 traite le signal de deux manières :
 - lorsque /BERR est affirmé seul, il exécute une exception de type erreur de bus après avoir terminé le cycle défaillant
 - lorsque /HALT est affirmé avant ou en même temps que /BERR, il réexécute le cycle de bus
- /BERR doit être désactivé au moins 1 cycle horloge avant la désactivation de /HALT

Exemple d'utilisation de l'entrée /BERR - montage de type chien de garde



4.5. Codes fonction

Sorties à trois états FC0, FC1, FC2

- spécifie le type de cycle de bus courant
 - accès code d'instruction ou donnée
 - acquiescement d'interruption
- FC2 :
 - mode Utilisateur ou Superviseur
 - même valeur que le bit S du registre d'état
- fournissent une protection basique de l'espace mémoire (actuellement, les MMU sont utilisés préférentiellement)

FC2	FC1	FC0	Cycle Type
0	0	0	-
0	0	1	User data
0	1	0	User program
0	1	1	-
1	0	0	-
1	0	1	Supervisor data
1	1	0	Supervisor program
1	1	1	Interrupt acknowledge

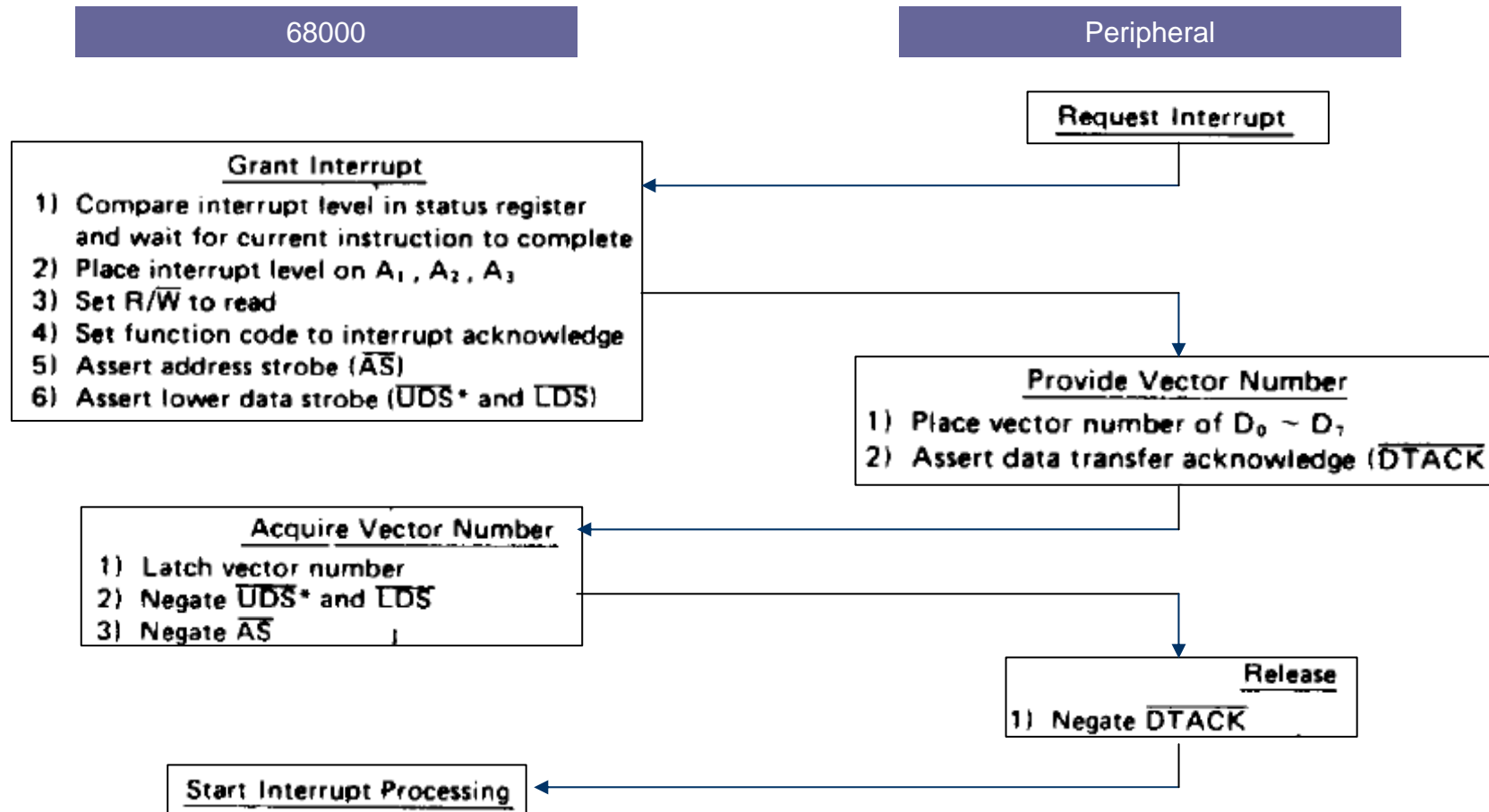
4.6. Signaux de contrôle des interruptions

/IPL0, /IPL1 et /IPL2 (Interrupt Priority Level)

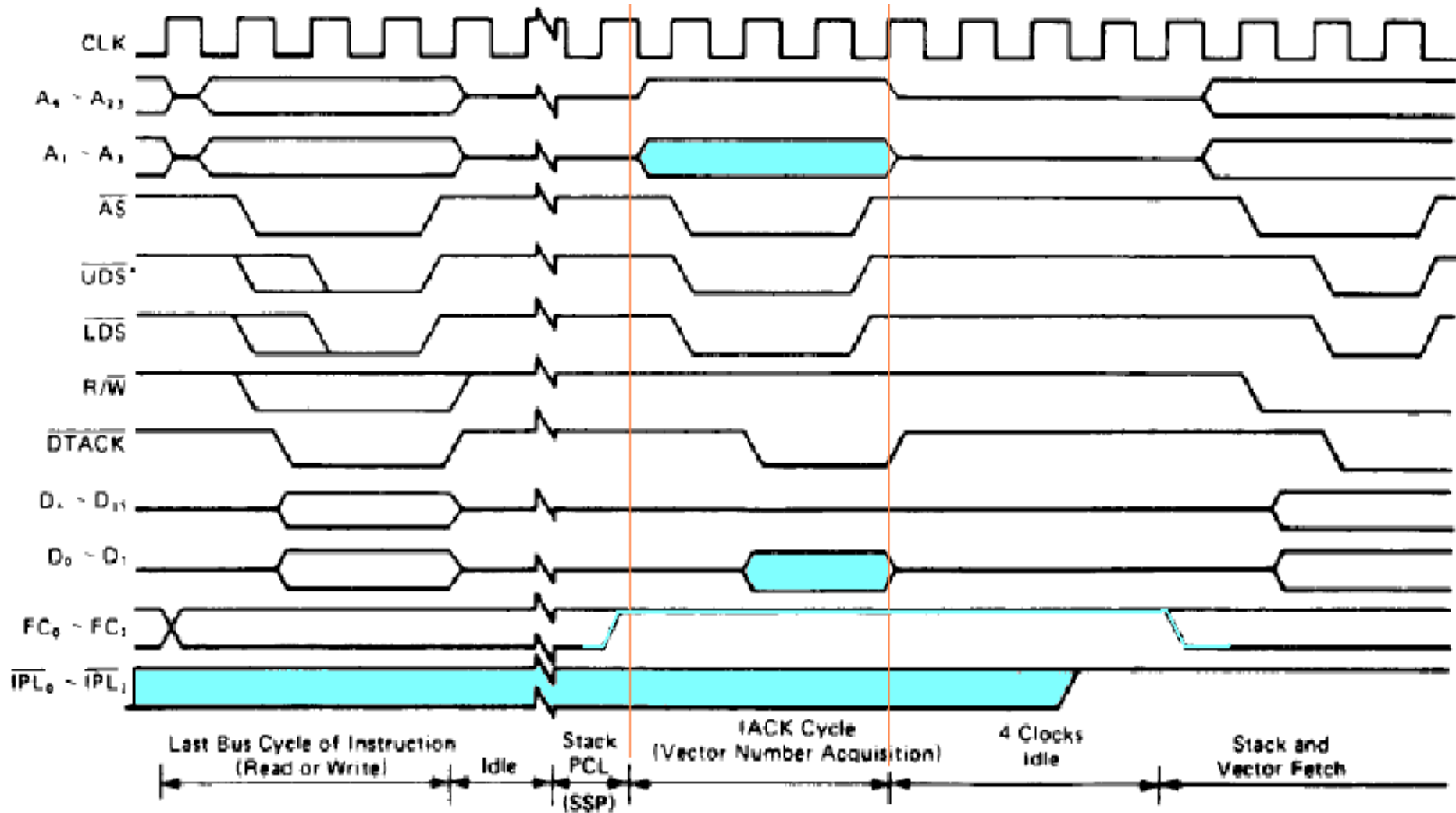
- entrées
- permet de suspendre (d'interrompre) temporairement l'exécution d'un programme pour traiter une tâche plus importante
- les circuits externes peuvent émettre une demande d'interruption au CPU en utilisant les entrées codées /IPL0, /IPL1 et /IPL2
- /IPL0, /IPL1 et /IPL2 permettent de coder 7 niveaux de priorité de demande d'interruption, de 1 à 7.
- les interruptions sont vectorisées ou autovectorisées (usage de /VPA – compatibilité avec les périphériques de génération 6800 – vecteur 25 à 31)
- une interruption peut être ignorée si son niveau est inférieur ou égal à celui du masque d'interruption I2, I1, I0 qui se situe dans le registre d'état.
- fonctionne avec FC0-FC2 au niveau du processus d'acquittement.

IPL2	IPL1	IPL0	Value	Meaning
H	H	H	000	Interrupt Level 0, No interrupt
H	H	L	001	Interrupt Level 1
H	L	H	010	Interrupt Level 2
H	L	L	011	Interrupt Level 3
L	H	H	100	Interrupt Level 4
L	H	L	101	Interrupt Level 5
L	L	H	110	Interrupt Level 6
L	L	L	111	Interrupt Level 7, Non-maskable

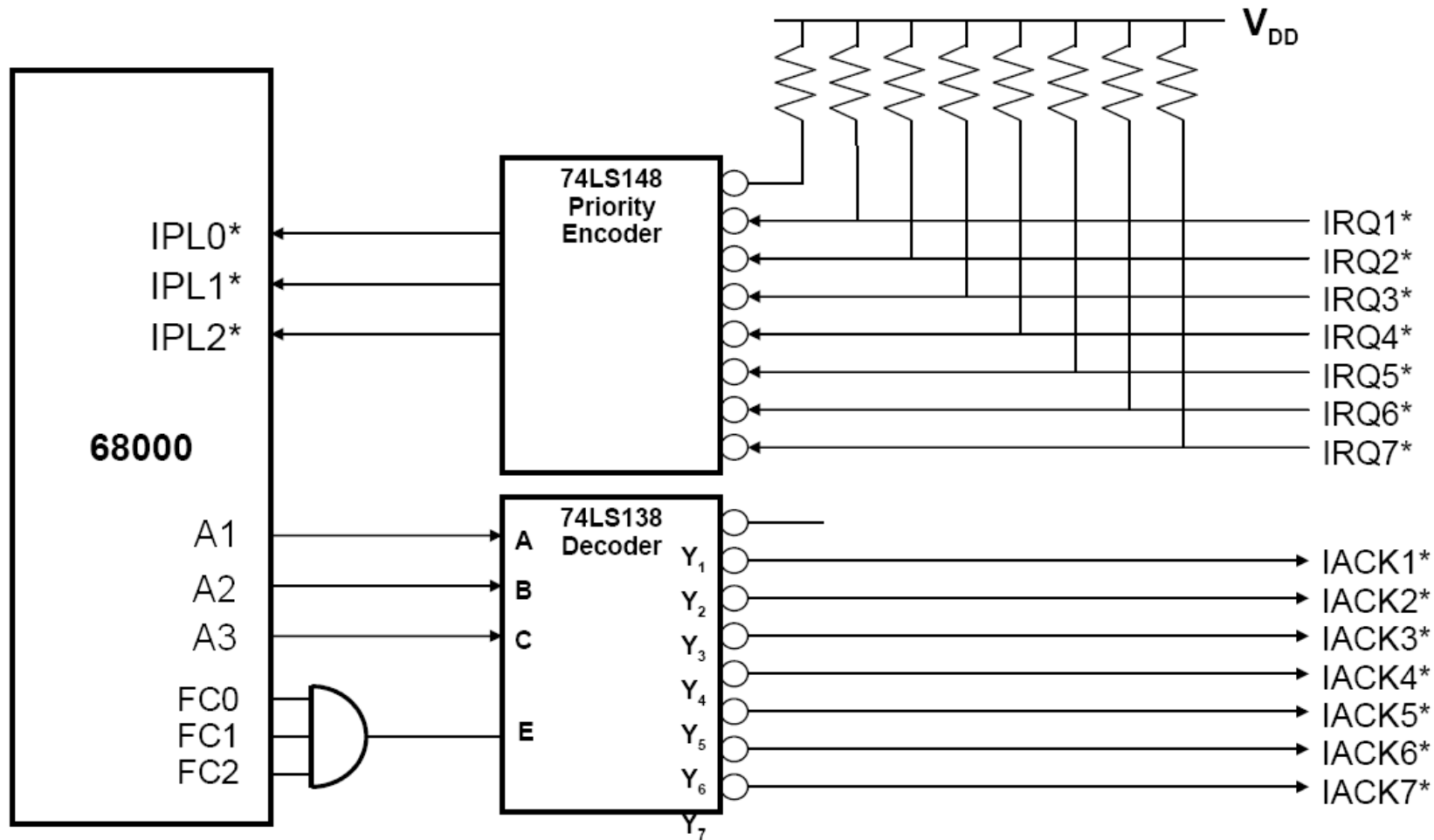
Séquencement des opérations lors d’une demande d’interruption



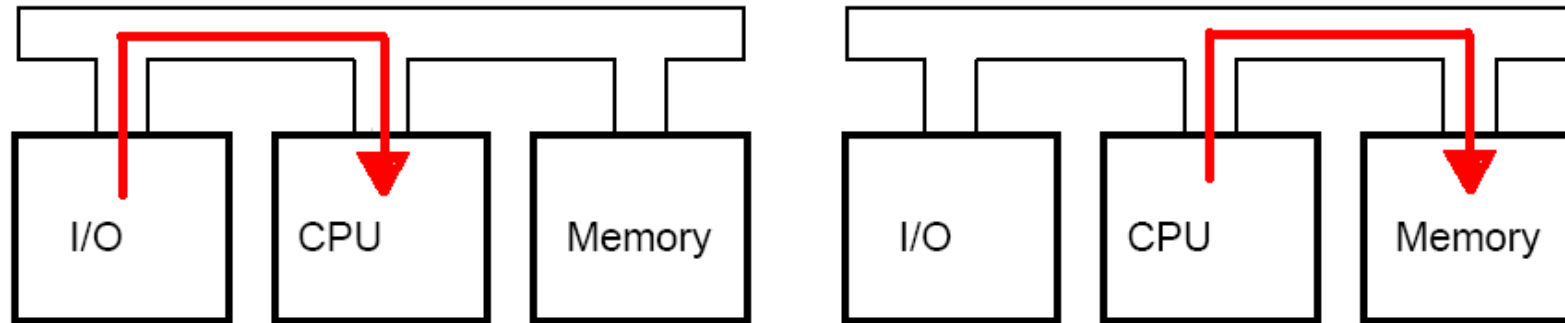
Chronogrammes d'acquiescement de demande d'interruption



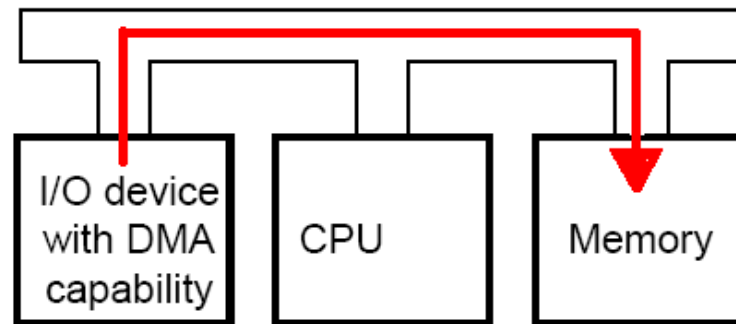
Exemple de génération du signal d'aquiescement de demande d'interruption



4.7. Signaux d'arbitrage du bus

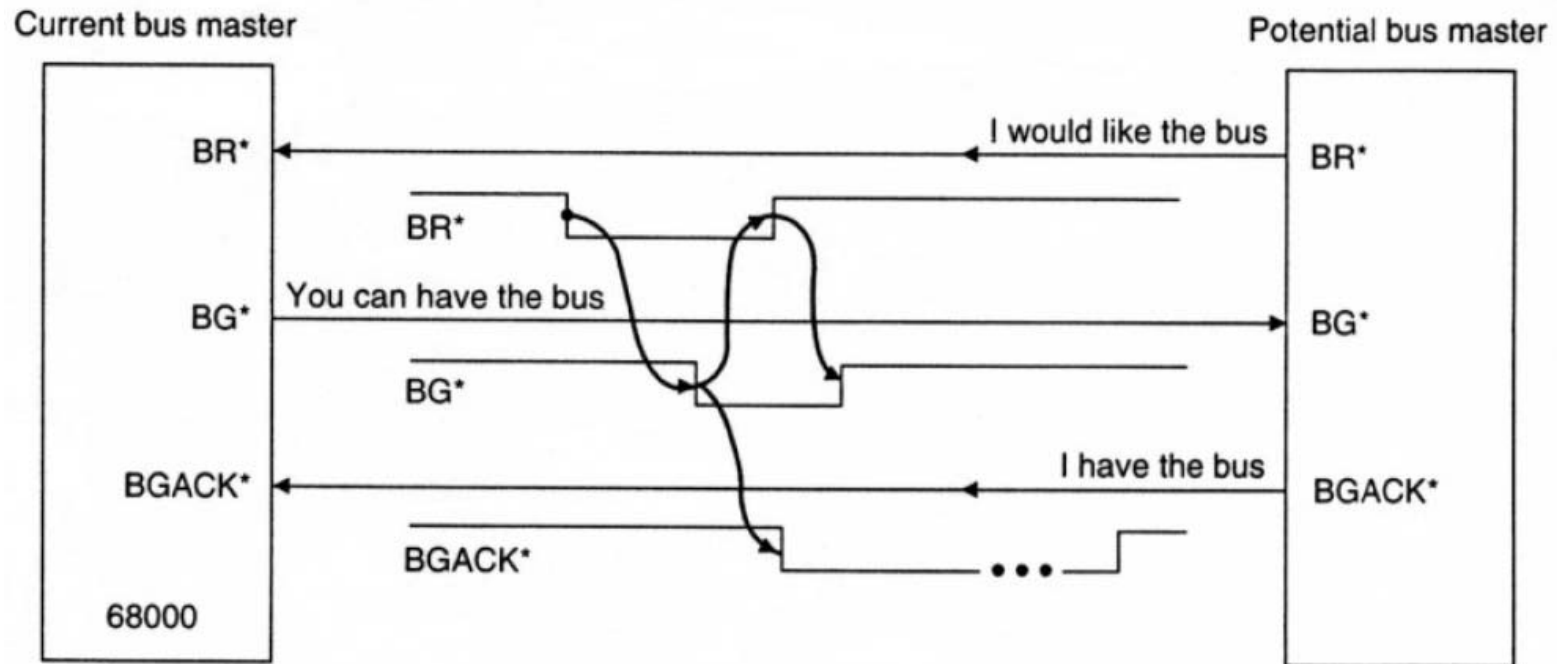


Without DMA: data is transferred in two steps



With DMA: data is sent directly to memory, saving a lot of time

Arbitrage du bus



- ce mécanisme est utilisé pour autoriser les microprocesseurs à gérer le DMA et le mutiprocasseur
- lorsque le 68000 contrôle le bus, il est maitre du bus. Ce rôle peut cependant être confié à un contrôleur DMA ou à un autre processeur.
- l'arbitrage consiste à désigner celui qui sera maitre du bus à la prochaine demande

4. MICROPROCESSEUR 68000 (MATERIEL) – SIGNAUX D'ECHANGES

- Signaux d'arbitrage du bus
 - /BR (bus request)
 - entrée drain ouvert – un OU câblé peut être réalisé avec d'autres circuits
 - affirmé par d'autres circuits voulant devenir maitre du bus
 - /BG (bus grant)
 - sortie
 - le maitre du bus lâche le contrôle de celui-ci à la fin de son cycle de bus
 - le demandeur peut négativer le signal /BR après l'affirmation de /BG
 - /BGACK (bus grant acknowledge)
 - entrée
 - le demandeur après l'affirmation de /BG affirme /BGACK pour informer qu'il est maintenant maitre du bus
 - /BGACK est maintenu à l'état bas tant que le nouveau maitre du bus reste maitre

Exemple

