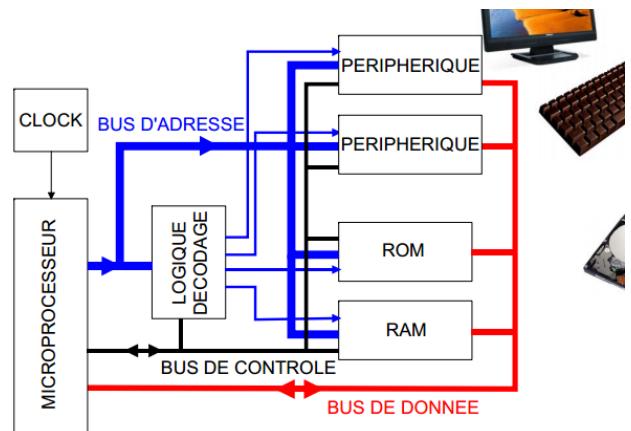


## Chapitre 3. Etude générale des interfaces d'entrées-sorties

## et Chapitre 4: Les échanges de données

### Introduction

Le microprocesseur est inutile en soi. Pour créer un système à base de microprocesseur viable, il faut des composants de mémoire, des composants d'interface, des circuits de synchronisation, de contrôle et une alimentation. La fonction d'un microprocesseur est le traitement de l'information. Le microprocesseur acquiert cette information et la restitue au moyen des unités d'entrées et de sorties (E/S ou input/output: I/O). Ces unités se nomment les périphériques car il se trouvent au voisinage du microprocesseur, à sa périphérie et non pas distant ou loin du microprocesseur tels que en entrée: clavier, souris et en sortie : imprimantes et écrans. Ces dispositifs matériels permettant d'assurer les échanges d'informations en entrée et en sortie entre le microprocesseur et l'extérieur ou de stocker de manière permanente des informations (voir figure). Généralement on parle de port d'E/S, qui est un connecteur à l'extrémité d'un bus dans lequel est connecté un périphérique.



Dans cette figure on note le bloc de l'horloge qui donne le signal d'horloge: CLK (4MHz, 6MHz, 8MHz) ; Le bloc d'alimentation utilise: Vcc=5V, GND=0V ; Les périphériques et les différents types de mémoire. L'ensemble est relié par un bus externe: donnée, adresse et contrôle.

### Les Interfaces d'entrée sortie

Les périphériques possèdent des propriétés très différentes tant du point de vue fonctionnalité que du point de vue électrique :

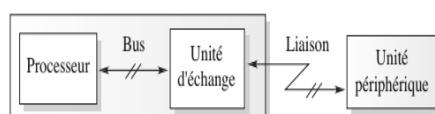
- Par exemple un clavier envoie les caractères à un rythme défini par l'utilisateur, chaque caractère étant lui-même codé et transmis avec une vitesse exprimée en bits par seconde ou bps ;
- Alors qu'un écran permet d'afficher des textes et/ou images en couleur. La transmission des informations entre unité centrale et moniteur peut se faire sous forme de train binaire, de signal composite ou vidéo, selon les caractéristiques du terminal utilisé ;

Le processeur communique avec l'extérieur par l'intermédiaire d'un bus qui relie les différents modules constitutifs de l'unité centrale. Les caractéristiques du bus ne permettent pas la connexion directe de périphériques. Ces derniers devront être "attachés" au bus de l'ordinateur à l'aide de dispositifs appelés unités d'échange, interfaces ou encore contrôleur d'échange

Le rôle des interfaces peut se limiter à une simple adaptation matérielle sur les signaux électriques ou bien aller jusqu'à la gestion de tout l'échange d'informations entre la partie processeur-mémoire et un périphérique. Plus précisément, on distingue :

1. L'adaptation matérielle :

- niveaux de tension ;
- mode de commande (en courant ou en tension) ;
- fourniture de puissance électrique ;
- mise en forme des signaux (niveaux, impulsions).



2. L'adaptation fonctionnelle :

- type de données (mot, octet, bloc...);
- vitesse de transfert ou débit ;
- gestion d'un protocole d'échange (ajonction et reconnaissance d'informations de service, prise en charge de toutes les procédures d'échange). Exemple une unité UART (pour transmission série asynchrone)

## Memory mapped I/O (mappage mémoire)

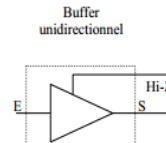
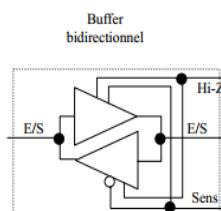
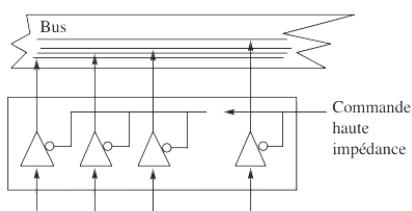
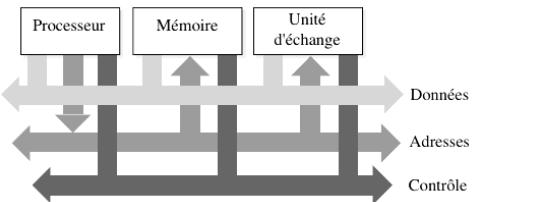
Le M68000 utilise des E / S mappés en mémoire (memory-mapped IO), ce qui signifie simplement qu'on peut adresser les ports périphériques comme s'il s'agissait d'adresses mémoire.

Dans le cas de MOTOROLA il n'y a pas d'instruction spéciale pour les entrées et sorties, au contraire de INTEL qui réserve des instructions exclusives pour faire une lecture IN et une écriture OUT.

Les organes d'E/S sont considérés comme des positions mémoire où l'on lit (Entrée) ou écrit (Sortie) des données. L'avantage de cette approche est que l'on peut utiliser le jeu d'instructions complet de microprocesseur pour manipuler les informations provenant de ces organes. L'inconvénient est que la zone d'E/S de petite dimension "emprise" sur l'espace total adressable normalement réservé à la mémoire.

Les interfaces d'E/S sont souvent des circuits intégrés spécialisés directement interfaçables avec le microprocesseur et qui assurent les liens avec les unités matérielles du système. Chaque interface dispose d'un champ adresse qui lui est propre. Ce champ adresse est fourni par le bloc de décodage d'adresse.

L'ensemble des bus de données, d'adresses et de contrôle constitue ce que l'on désignera par bus externe. Sur ce bus peuvent être connectés plusieurs processeurs ou unités d'échange susceptibles de prendre le contrôle des bus - systèmes multiprocesseurs ou dispositifs d'accès direct à la mémoire (DMA). Il est donc nécessaire de protéger les autres unités connectées (pour éviter les courts circuits, une unité place un 1(5V) et une autre 0 v sur le bus ce qui endommagera la ligne d'où la nécessité d'un troisième état (circuit ouvert)). Ceci est réalisé à l'aide de tampons trois états (Three States Drivers) pouvant posséder les états 0, 1 et haute impédance. Ce dernier état équivaut à une déconnexion physique de la ligne.



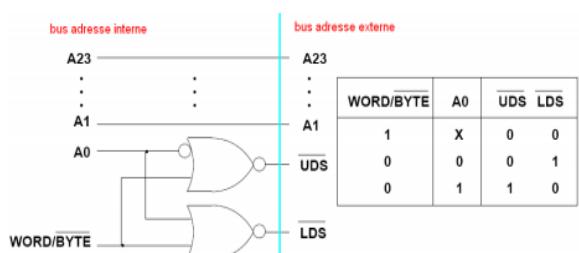
## Signaux du 68000

Les différents signaux issus du MC 68000 sont cités avec leur fonction dans ce qui suit :

Bus de données : Bus bidirectionnel trois états de 16 bits.

Bus d'adresses : Bus unidirectionnel de 23 bits sans bit A0. La figure suivante montre la génération de la ligne A0 à partir des lignes UDS et LDS pour en faire un bus d'adresse externe de 24 bits

Bus de contrôle : pour les échanges : Il faut remarquer que la plus part des lignes de contrôle sont activées bas autrement dit c'est l'état 0 qui fait activer la ligne alors que l'état 1 est considéré comme l'état de repos. Ceci est indiqué par une barre sur le nom de ligne de contrôle exemple : AS (voir tableau ci-dessous : tristate : 3ième état)



CLK : horloge

AS : Adresse Strobe

UDS LDS : Upper/Lower Data Strobe

R/W : Read/Write

Echange asynchrone : DTACK : DaTa ACKnowledge

Echange synchrone : E : Enable (horloge) VMA : Valid Memory Adress VPA : Valid Peripheral Address

Etat du processeur décrit par FC0, FC1, FC2

Traitement des exceptions :

RES : RESet

HLT : HALT. En entrée, stoppe le processeur. En sortie, indique un arrêt du processeur après une double erreur de bus.

BERR : Bus ERROR. Détection d'une erreur de bus

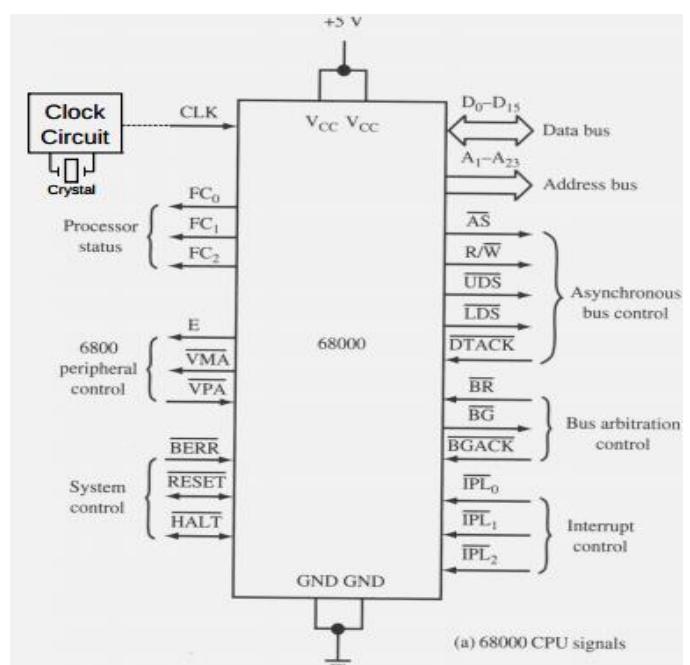
IPL 1 2 3 : Interrupt Pending Level. Interruptions externes.

Gestion d'attribution des bus : BR : Bus Request. Indique qu'un système extérieur désire prendre le contrôle des bus.

BG : Bus Grand. Indique que le processeur va libérer les bus à la fin du cycle en cours.

BGACK : Bus Grand ACKnowledge. Indique qu'un autre circuit a pris le contrôle des bus.

Signal	Input	Output	Tristate
CLK	✓		
FC <sub>0</sub> -FC <sub>2</sub>		✓	✓
E		✓	
VMA		✓	✓
VPA	✓		
BERR	✓		
RESET	✓	✓	
HALT	✓	✓	
IPL <sub>0</sub> -IPL <sub>2</sub>	✓		
BR	✓		
BG		✓	
BGACK	✓		
AS		✓	✓
R/W		✓	✓
UDS		✓	✓
LDS		✓	✓
DTACK	✓		
A <sub>1</sub> -A <sub>23</sub>		✓	✓
D <sub>0</sub> -D <sub>15</sub>	✓	✓	✓



### Gestion des échanges

Les données sont des variables ou des instructions. Si leur intégrité n'est pas assurée, le système risque de planter. Il existe deux manières d'assurer l'intégrité des échanges :

Synchrone : L'intégrité de la transmission est assurée en synchronisme (cohérence) avec une horloge.

Intérêt : Le récepteur est synchronisé sur l'émetteur des données.

Inconvénient : Il n'y a pas de retour ; l'émetteur ne sait pas si la donnée a bien été transmise.

Asynchrone : L'intégrité de la transmission est assurée par des signaux de validation.

Ces deux types d'échange existent sur le 68000.

**Echange asynchrone** C'est le mode d'échange par défaut du 68000. Lorsqu'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'acquittement qui indique au CPU que le cycle de bus actuel est complété. La séquence des opérations est appelée poignée de mains (handshake).

### Signaux mis en œuvre

/AS (sortie) • affirmé pour indiquer qu'une adresse valide est placée sur le bus d'adresse /UDS and /LDS • sortie à 3 états • détermine le format (taille) de la donnée échangée :

DTACK • entrée (le seul signal de contrôle en entrée) • signal de poignée de mains (handshake) générée 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 • sortie à 3 états • spécifie la nature du cycle de bus • R/W = 1 → lecture • R/W = 0 → 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

#### Cycle d'écriture d'un mot

Processeur

Place l'adresse

Place les codes fonction Fc0, Fc1, Fc2

Affirme l'échantillonnage d'adresses AS

Etablit R/W / en écriture

Place les données

Affirme l'échantillonnage des données

UDS LDS

Esclave

Décode les adresses

Lit les données

Affirme la reconnaissance de transfert des données DTACK

Processeur

Infirme UDS LDS,

Infirme AS

Rétablissement R/W / en lecture (état haut)

Fin de cycle

Esclave

Infirme DTACK

Pour l'écriture d'un octet, seuls UDS, LDS, changent suivant qu'il s'agit d'un octet de poids fort ou de poids faible.

#### Cycle de lecture d'un mot

Processeur

Etablit R/W / en écriture

Place l'adresse

Place les codes fonction Fc0, Fc1, Fc2

Affirme l'échantillonnage d'adresses AS

Affirme l'échantillonnage des données

UDS LDS

Lit les données

Infirme UDS, LDS,

Infirme AS

Fin de cycle

Esclave

Décode les adresses

Ecrit les données

Affirme la reconnaissance de transfert des données DTACK

Esclave

Infirme DTACK

#### Echange synchrone

Permet au 68000 d'utiliser des circuits de la précédente génération 6800. Les échanges synchrones existent sur le 68000 pour le rendre compatible avec les circuits de la famille 6800 : exemple de ces circuits :

6821 : Port parallèle d'E/S

6840 : Timer

6850 : ACIA (liaison série)

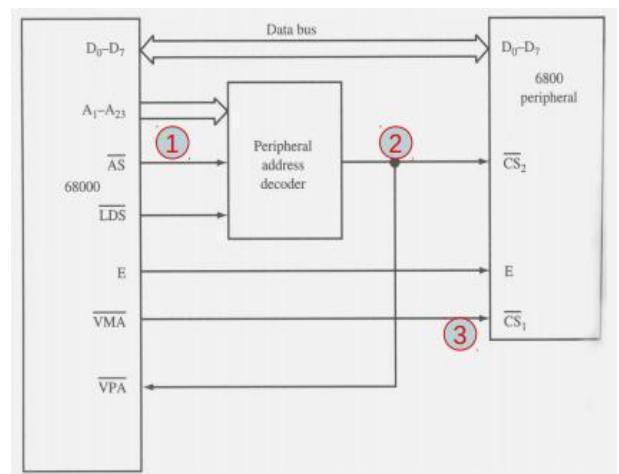
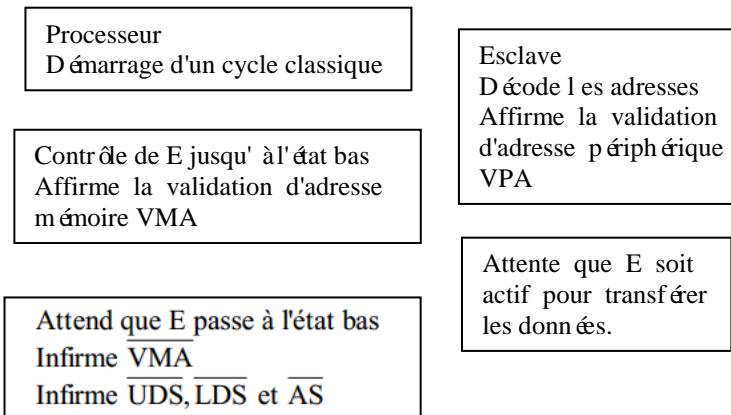
#### Signaux mis en œuvre

- 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

Comme pour certains composants asynchrones, on peut être amené à générer le signal VPA à l'aide d'une logique externe

Remarque : Les composants de la famille 6800 peuvent tourner jusqu'à 1Mhz et le 68000 jusqu'à 10Mhz. Pour que les temps de transferts soient compatibles, E est 10 fois moins rapide que CLK.

#### Echange synchrone : Différents cycles de lecture et écriture



#### Les modes d'échange

Lorsque le processeur exécute un programme, les bus d'adresses, de données et de contrôle voient leur état évoluer au cours du temps. L'exécution d'une opération d'écriture ou de lecture (en mémoire ou sur d'autres dispositifs) se fait par présentation :

- sur le bus de données : de la valeur de la donnée que l'on veut transférer,
- et sur le bus d'adresses : de l'adresse de l'unité de sortie ou d'entrée.

Un échange de données entre un organe d'E/S et un périphérique nécessite en général 2 lignes de dialogue: l'une indiquant la transmission d'une donnée (validation) par l'émetteur, l'autre indiquant que le récepteur a pris en compte cette donnée (acquittement)

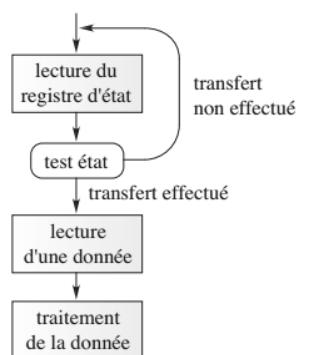
Le transfert de données entre unités d'échange et processeur fait intervenir trois techniques :

1. interrogation ou polling (scrutiny) ;
2. accès direct à la mémoire ou ADM (DMA ou Direct Memory Access) ;
3. interruptions.

#### Le mode d'échange par interrogation

Ce mode est caractérisé par le fait que le processeur interroge en permanence les unités d'échange. Il vient lire les registres d'état et teste les bits concernés par l'échange. À la suite de quoi, la ou les données reçues sont lues ou écrites.

Ce mode de travail, aussi désigné par le terme de polling, test d'état ou attente active, est particulièrement simple à mettre en œuvre mais présente l'inconvénient de mobiliser toute l'activité du processeur pour l'opération d'entrée-sortie. On le trouve le plus souvent implanté sur des matériels du type micro-ordinateurs dotés



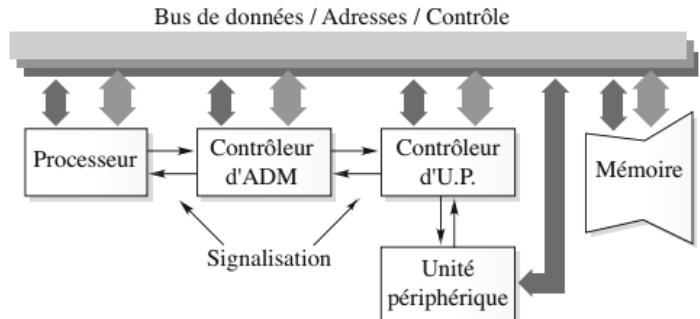
d'un système d'exploitation rudimentaire.

### L'accès direct à la mémoire

L'accès direct à la mémoire (ADM) (DMA ou Direct Memory Access) consiste à donner à une unité d'échange la possibilité d'aller lire ou écrire directement dans la mémoire sans passer par le processeur. Cette méthode permet :

- De soulager le processeur de ce travail de transfert,
- Éventuellement, de remplacer le processeur dans le cas où celui-ci n'est pas assez rapide pour effectuer le transfert processeur - unité d'échange entre deux transferts unité d'échange ↔ unité périphérique

Le contrôleur d'ADM s'intercale entre le processeur et le contrôleur de l'unité périphérique.



### Entrées-sorties pilotées par les interruptions

Une interruption est un événement externe provoquant l'arrêt du travail courant du processeur en vue d'exécuter une autre tâche. Cet événement se traduit physiquement par une variation de tension sur une ligne "entrant" dans l'unité de contrôle du processeur. Le périphérique utilise le mécanisme des interruptions pour signaler qu'il est prêt.

Les demandes d'interruption sont effectuées en appliquant un ou plusieurs états actifs sur les broches IPL2, IPL1, IPL0. L'interruption est alors pendante (en instance). La demande ne sera satisfaite qu'après l'exécution de l'instruction en cours. Si le niveau de priorité de la demande est strictement supérieur au masque d'interruption situé dans le registre d'état celle-ci sera acceptée par le processeur. Une exception est faite pour l'interruption de niveau 7 qui ne peut être masquée, elle sera toujours acceptée quelle que soit la valeur introduite dans le masque d'interruption, c'est une interruption non masquable. Une copie interne de SR (registre d'état) est effectuée, le processeur est mis en l'état superviseur, le mode Trace est supprimé, le masque d'interruption est mis au niveau de l'interruption prise en compte. Le processeur cherche le N° de vecteur du circuit interrompant en effectuant une lecture alors que les broches de code fonction (FC2, FC1, FC0) sont à 111, R/W à 1, AS, LDS, UDS à 0, A5 et A4 à 1 et les trois bits d'adresse A3, A2, A1 positionnés au niveau de l'interruption en cours de traitement.

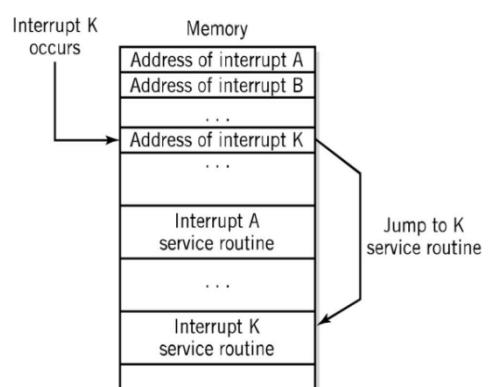
Un circuit de décodage devra exploiter ces informations de telle sorte que plaçant un état actif sur la ligne IACK du circuit interrompant (cas d'un périphérique 68000) celui-ci fournit le N° vecteur (de 64 à 255) sur les 8 lignes de poids faible du bus de données et tire la ligne DTACK du processeur à 0.

Si le N° Vecteur n'est pas générée par le circuit interrompant, une logique externe devra activer la ligne VPA du 68000 pour en demander la génération interne. Les N° fournis iront alors de 25 pour le niveau 1, à 31 pour le niveau 7.

Lorsque le N° vecteur est obtenu, le processeur sauve PC et SR sur la pile superviseur. La valeur de PC empilée est l'adresse de l'instruction qui aurait été exécutée en l'absence d'interruption.

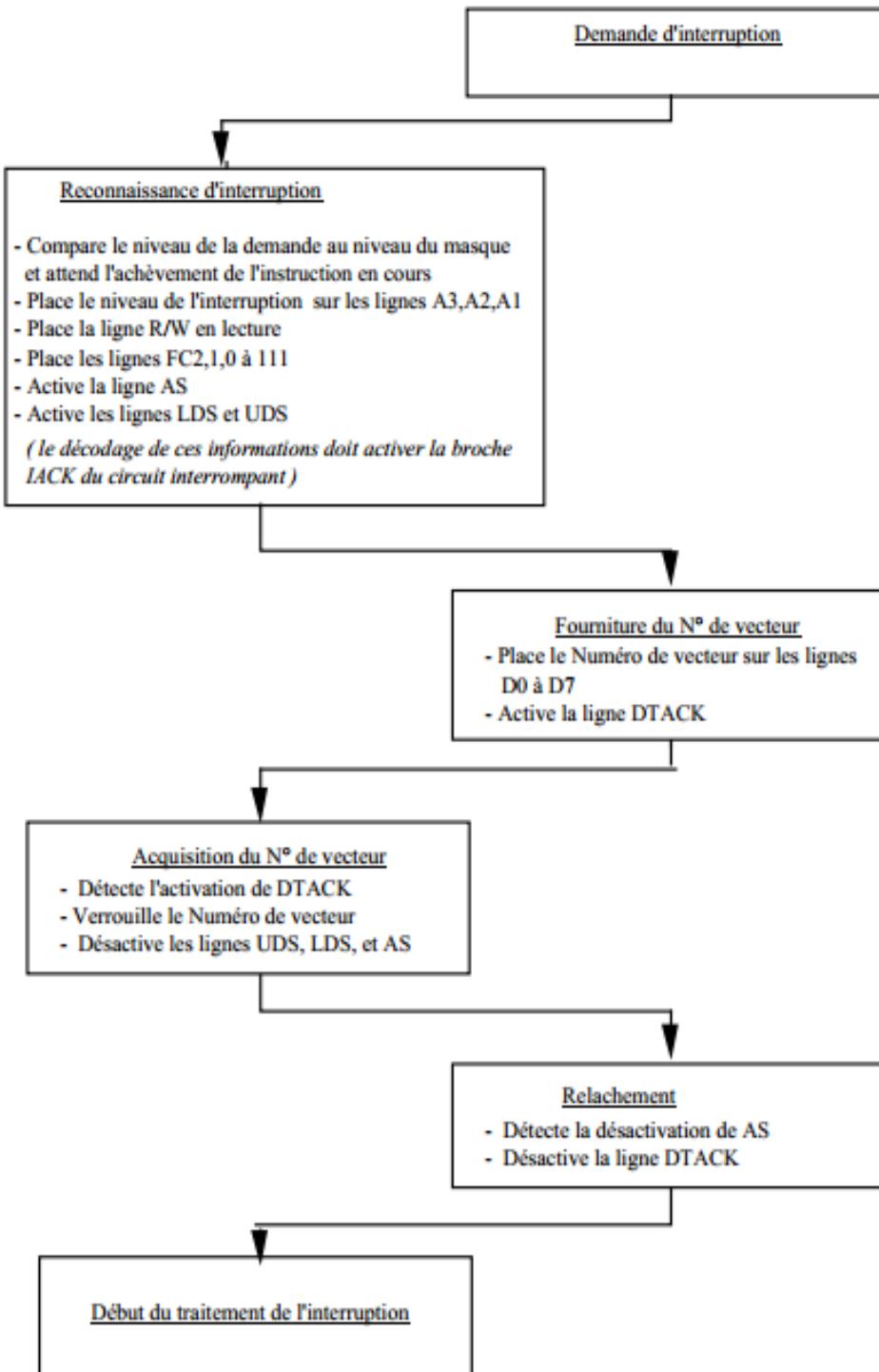
Toutes les interruptions sont traitées en mode superviseur et utilisent la pile pointée par SSP quel que soit l'état du bit S avant l'entrée en exception.

Le programme de traitement d'interruption se termine par une instruction spéciale de retour d'interruption (REI, RTI, ...) qui se charge de récupérer dans la pile l'adresse de retour et, éventuellement, le contenu d'autres registres. Cette instruction est semblable à l'instruction retour de sous-programme (RET, RTS).



## MICROPROCESSEUR

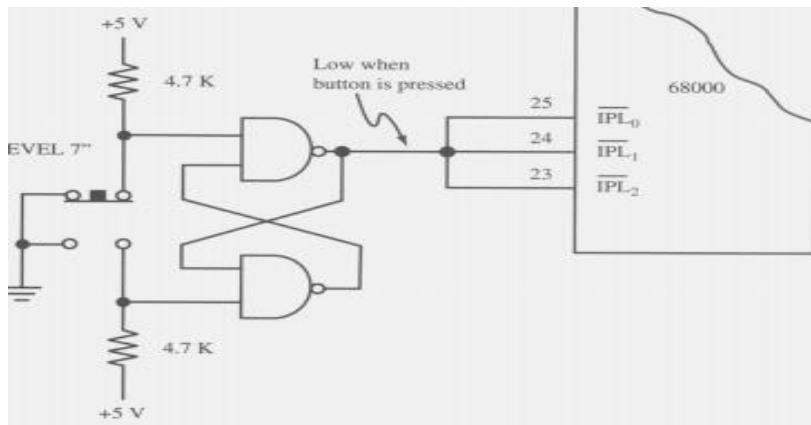
## CIRCUIT INTERROMPANT



### Échanges entre le circuit interrompant et le microprocesseur

- /IPL0, /IPL1 et /IPL2 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 • 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-FC1-FC2 au niveau du processus d'acquittement

### Exemple de matériel: génération d'une interruption de niveau 7 par un bouton poussoir anti rebond



Quand on appuie sur le bouton on genere un niveau bas aux entrées IPL2,IPL1,IPL0 du microprocesseur qui correspond à une interruption de niveau 7

## Accès direct à la mémoire

Mécanisme d'interruptions est efficace

- Toutefois il ne faut pas que le temps utilisé par le processeur pour le programme de gestion d'interruption et du pilote soit trop important. Solution:
  - Diminuer le nombre d'interruptions
  - Chargement à partir de la mémoire principale sans utilisation du processeur central
  - Dispositif DMA (Direct Memory Access)
    - Composant matériel comprend
      - Un registre d'adresse
      - Un registre de comptage
      - Un registre de commande (lecture ou écriture)
      - Une zone tampon permettant le stockage de données
      - Un composant actif, de type processeur

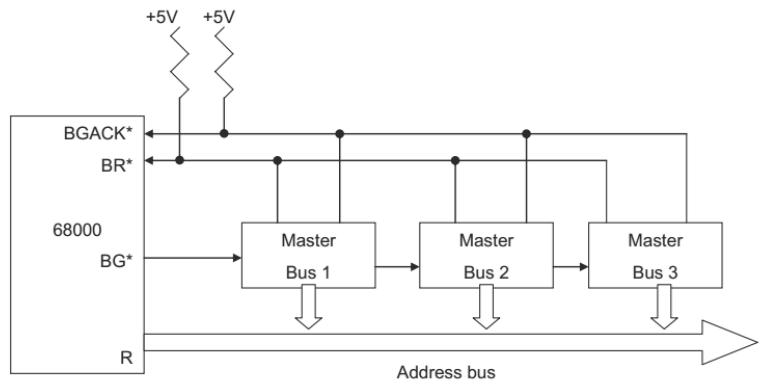
### Arbitrage du bus

- ce mécanisme est utilisé pour autoriser les microprocesseurs à gérer le DMA et le multiprocesseur
- lorsque le 68000 contrôle le bus, il est maître 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 maître du bus à la prochaine demande

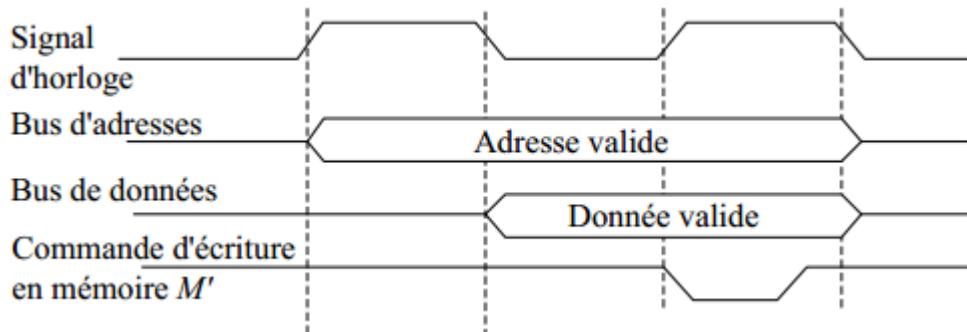
### Signaux d'arbitrage du bus

- /BR (bus request) • entrée drain ouvert – un OU câblé peut être réalisé avec d'autres circuits
- affirmé par d'autres circuits voulant devenir maître du bus
- /BG (bus grant) • sortie
- le maître du bus lâche le contrôle de celui-ci à la fin de son cycle de bus
- le demandeur peut inactiver 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 maître du bus
- /BGACK est maintenu à l'état bas tant que le nouveau maître du bus reste maître



## REMARQUE

Un diagramme des temps d'une instruction d'écriture est généralement donné pour expliquer l'échange de données en fonction du temps. On y trouve l'horloge du processeur, le bus de données, le bus d'adresse et la ligne d'écriture du bus de contrôle. Cette dernière sert à valider toute opération d'écriture engendrée par l'exécution d'une instruction d'écriture (en mémoire ou vers un autre dispositif de sortie) exemple voir figure ci dessous.



Les adresses sont utilisées pour sélectionner l'unité d'entrée-sortie choisie

