

VI - 3 - Temps d'exécution des instructions

VI - 3 - 1 - Mécanisme de recherche des instructions

Le 68000 utilise le mécanisme de “**prefetch**” (pré-recherche) des instructions. Il recherche systématiquement deux mots consécutifs dans la mémoire de programme.

Dans les explications qui vont suivre, on considèrera que l'exécution d'une instruction commence lorsque le microprogramme est activé, c'est à dire lorsqu'il y a réellement début de l'exécution.

Il est difficile avec ce type de microprocesseur de déterminer avec précision le temps d'exécution d'une suite d'instructions notamment en raison des éléments suivants:

- Lorsque l'exécution d'une instruction est commencée, il est fréquent que le mot instruction de la suivante ait déjà été recherché et se trouve dans le décodeur. L'exécution d'une instruction et le décodage de la suivante peuvent être simultanés, il en résulte un gain de temps.
- La dernière recherche dans une instruction multimots est faite lorsque le mot opération est abandonné et que le décodage de la suivante commence.
- Si l'instruction tient en un mot (le premier d'une recherche) causant un branchement, le second mot n'est pas utilisé, il est possible alors de constater une recherche superflue.
- En cas d'interruption ou d'exception, il est possible que deux mots ne soient pas utilisés.

VI - 3 - 2 - Temps d'exécution

Les tableaux qui suivent vont permettre le calcul des temps d'exécution exprimés en cycles d'horloge.

Il n'est pas tenu compte de cycles d'attente éventuels qui pourraient être dus à des temps d'accès trop importants des circuits périphériques.

Chacun des cycles d'écriture ou de lecture en mémoire est compté pour 4 cycles d'horloge.

Le calcul du temps d'exécution d'une instruction nécessitera l'utilisation de **deux tableaux**, l'un donnant le temps d'obtention des opérandes (c'est à dire recherche dans la mémoire de programme des éléments nécessaires, calcul éventuel de l'adresse et lecture de l'opérande), et l'autre le temps d'exécution proprement dit c'est à dire la recherche de l'instruction, l'exécution et le rangement du résultat.

Les nombres qui apparaissent dans ces tableaux indiquent le nombre de cycles d'horloge nécessaires pour le travail à effectuer et, entre parenthèses, le nombre de lectures et le nombre d'écritures nécessaires à ce travail.

1er Exemple:

Temps d'exécution de **ADD.W 0X00100000,D0**
ceci est de la forme **ADD.W <AE>,D0**

Temps recherche et d'exécution de **ADD.W <AE>,D0** **4 (1/0)**
(constitué essentiellement par la recherche du code instruction en mémoire)

Temps de calcul et de recherche de l'opérande **<AE>**, adressage absolu long **12 (3/0)**
(deux lectures en mémoire pour acquérir le mot long adresse de l'opérande source et lecture de l'opérande source)

Soit 4 + 12 = 16 cycles d'horloge , 4 accès mémoire en lecture, 0 en écriture

2ème Exemple:

Temps d'exécution de **ADD.W D0,0X00100000**
ceci est de la forme **ADD.W Dn,<AE>**

Temps d'exécution de **ADD.W D0,<M>** **8 (1/1)**
(constitué essentiellement par la recherche du code instruction en mémoire et écriture du résultat à l'adresse destination)

Temps de calcul et recherche de **<AE>**, adressage absolu long **12 (3/0)**
(deux lectures en mémoire pour acquérir le mot long adresse de l'opérande destination et lecture de l'opérande destination)

Soit 8 + 12 = 20 cycles d'horloge , 4 accès mémoire en lecture, 1 en écriture

3ème Exemple:

Temps d'exécution de **ADD.W D0,D1**
 ceci est de la forme **ADD.W <AE>,Dn**

Temps d'exécution de **ADD.W <AE>,Dn** **4 (1/0)**
 (Lecture du mot instruction en mémoire)
 Temps de calcul de <AE>, adressage direct **0 (0/0)**
 (pas d'opération externe la lecture ou d'écriture opérations essentiellement internes)
Soit 4 cycles d'horloge , 1 accès mémoire en lecture, 0 en écriture

VI - 3 - 3 - Spécifications**VI - 3 - 3 - 1 - Obtention d'un opérande**

Le tableau ci-dessous donne le temps nécessaire au microprocesseur pour lire l'adresse ou l'information adresse en mémoire de programme, élaborer l'adresse de l'opérande et rechercher l'opérande.

MODES D'ADRESSAGE		Opérande Octet, Mot		Opérande Mot Long	
Dn	Registre de données direct	0	(0/0)	0	(0/0)
An	Registre d'adresses direct	0	(0/0)	0	(0/0)
(An)	Indirect par registre d'adresses	4	(1/0)	8	(2/0)
(An)+	Indirect par registre d'adresses post incrémenté	4	(1/0)	8	(2/0)
-(An)	Indirect par registre d'adresses pré décrémenté	6	(1/0)	10	(2/0)
d(An)	Indirect par registre d'adresses avec déplacement	8	(2/0)	12	(3/0)
d(An,Rx)*	Indirect par registre d'adresses avec déplacement et index	10	(2/0)	14	(3/0)
xxx.W	Absolu court	8	(2/0)	12	(3/0)
xxx.L	Absolu long	12	(3/0)	16	(4/0)
d(PC)	Indirect par PC avec déplacement	8	(2/0)	12	(3/0)
d(PC,Rx)*	Indirect par PC avec déplacement et index	10	(2/0)	14	(3/0)
#xxx	Immédiat	4	(1/0)	8	(2/0)

- La taille de l'index est sans influence sur le temps d'obtention de l'adresse
-

VI - 3 - 3 - 2 - Exécution des instructions

VI - 3 - 3 - 2 - 1 - Exécution des instructions MOVE sur des mots et octets

Source ↓	Destination								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,Rx)	xxx.W	xxx.L
Dn	4 (1/0)	4 (1/0)	8 (1/1)	8 (1/1)	8 (1/1)	12 (2/1)	14 (2/1)	12 (2/1)	16 (3/1)
An	4 (1/0)	4 (1/0)	8 (1/1)	8 (1/1)	8 (1/1)	12 (2/1)	14 (2/1)	12 (2/1)	16 (3/1)
(An)	8 (2/0)	8 (2/0)	12 (2/1)	12 (2/1)	12 (2/1)	16 (3/1)	18 (3/1)	16 (3/1)	20 (4/1)
(An)+	8 (2/0)	8 (2/0)	12 (2/1)	12 (2/1)	12 (2/1)	16 (3/1)	18 (3/1)	16 (3/1)	20 (4/1)
-(An)	10 (2/0)	10 (2/0)	14 (2/1)	14 (2/1)	14 (2/1)	18 (3/1)	20 (3/1)	18 (3/1)	22 (4/1)
d(An)	12 (3/0)	12 (3/0)	16 (3/1)	16 (3/1)	16 (3/1)	20 (4/1)	22 (4/1)	20 (4/1)	24 (5/1)
d(An,Rx)	14 (3/0)	14 (3/0)	18 (3/1)	18 (3/1)	18 (3/1)	22 (4/1)	24 (4/1)	22 (4/1)	26 (5/1)
xxx.W	12 (3/0)	12 (3/0)	16 (3/1)	16 (3/1)	16 (3/1)	20 (4/1)	22 (4/1)	20 (4/1)	24 (5/1)
xxx.L	16 (4/0)	16 (4/0)	20 (4/1)	20 (4/1)	20 (4/1)	24 (5/1)	26 (5/1)	24 (5/1)	28 (6/1)
d(PC)	12 (3/0)	12 (3/0)	16 (3/1)	16 (3/1)	16 (3/1)	20 (4/1)	22 (4/1)	20 (4/1)	24 (5/1)
d(PC,Rx)	14 (3/0)	14 (3/0)	18 (3/1)	18 (3/1)	18 (3/1)	22 (4/1)	24 (4/1)	22 (4/1)	26 (5/1)
#xxx	8 (2/0)	8 (2/0)	12 (2/1)	12 (2/1)	12 (2/1)	16 (3/1)	18 (3/1)	16 (3/1)	20 (4/1)

VI - 3 - 3 - 2 - 2 - Exécution des instructions MOVE sur des mots longs

Source ↓	Destination								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,Rx)	xxx.W	xxx.L
Dn	4 (1/0)	4 (1/0)	12 (1/2)	12 (1/2)	12 (1/2)	16 (2/2)	18 (2/2)	16 (2/2)	20 (3/2)
An	4 (1/0)	4 (1/0)	12 (1/2)	12 (1/2)	12 (1/2)	16 (2/2)	18 (2/2)	16 (2/2)	20 (3/2)
(An)	12 (3/0)	12 (3/0)	20 (3/2)	20 (3/2)	20 (3/2)	24 (4/2)	26 (4/2)	24 (4/2)	28 (5/2)
(An)+	12 (3/0)	12 (3/0)	20 (3/2)	20 (3/2)	20 (3/2)	24 (4/2)	26 (4/2)	24 (4/2)	28 (5/2)
-(An)	14 (3/0)	14 (3/0)	22 (3/2)	22 (3/2)	22 (3/2)	26 (4/2)	28 (4/2)	26 (4/2)	30 (5/2)
d(An)	16 (4/0)	16 (4/0)	24 (4/2)	24 (4/2)	24 (4/2)	28 (5/2)	30 (5/2)	28 (5/2)	32 (6/2)
d(An,Rx)	18 (4/0)	18 (4/0)	26 (4/2)	26 (4/2)	26 (4/2)	30 (5/2)	32 (5/2)	30 (5/2)	34 (6/2)
xxx.W	16 (4/0)	16 (4/0)	24 (4/2)	24 (4/2)	24 (4/2)	28 (5/2)	30 (5/2)	28 (5/2)	32 (6/2)
xxx.L	20 (5/0)	20 (5/0)	28 (5/2)	28 (5/2)	28 (5/2)	32 (6/2)	34 (6/2)	32 (6/2)	36 (7/2)
d(PC)	16 (4/0)	16 (4/0)	24 (4/2)	24 (4/2)	24 (4/2)	28 (5/2)	30 (5/2)	28 (5/2)	32 (6/2)
d(PC,Rx)	18 (4/0)	18 (4/0)	26 (4/2)	26 (4/2)	26 (4/2)	30 (5/2)	32 (5/2)	30 (5/2)	34 (6/2)
#xxx	12 (3/0)	12 (3/0)	20 (3/2)	20 (3/2)	20 (3/2)	24 (4/2)	26 (4/2)	24 (4/2)	28 (5/2)

VI - 3 - 3 - 2 - 3 - Exécution des instructions standard

Instruction	Taille	Opération <AE>,An	Opération <AE>,Dn	Opération Dn, <M>
ADD	Octet,Mot	8 (1/0)+	4 (1/0)+	8 (1/1)+
ADD	Mot Long	6 (1/0)+**	6 (1/0)+**	12 (1/2)+
AND	Octet,Mot		4 (1/0)+	8 (1/1)+
AND	Mot Long		6 (1/0)+**	12 (1/2)+
CMP	Octet,Mot	6 (1/0)+	4 (1/0)+	
CMP	Mot Long	6 (1/0)+	6 (1/0)+*	
DIVS			158 (1/0)+*	
DIVU			140 (1/0)+*	
EOR	Octet,Mot		4 (1/0)+***	8 (1/1)+
EOR	Mot Long		8 (1/0)+***	12 (1/2)+
MULS			70 (1/0)+*	
MULU			70 (1/0)+*	
OR	Octet,Mot		4 (1/0)+	8
OR	Mot Long		6 (1/0)+**	12
SUB	Octet,Mot	8 (1/0)+	4 (1/0)+	8 (1/1)+
SUB	Mot Long	6 (1/0)+**	6 (1/0)+**	12 (1/2)+

NOTES:

- + Ajouter le temps d'obtention de l'adresse
- * Indique une valeur maximum
- ** Le temps de base est 6 périodes d'horloge + 8 si AE est registre direct ou immédiat.
- *** Valable uniquement si AE est registre direct

DIVS,DIVU L'algorithme utilisé crée une variation inf à 10% entre le cas le plus et le moins favorable

MULU,MULS La multiplication demande $38 + 2n$ périodes d'horloge

MULU n est le nombre de 1 dans EA

MULS Ajoute un 0 à droite, n est le nombre de 10 et de 01 dans les 17 bits le cas le plus défavorable est 0X5555

VI - 3 - 3 - 2 - 4 - Exécution des instructions immédiates

Instruction	Taille	Opération #,An	Opération #,Dn	Opération #,<M>
ADDI	Octet,Mot		8 (2/0)	12 (2/1)+
ADDI	Mot Long		16 (3/0)	20 (3/2)+
ADDQ	Octet,Mot	8 (1/0)	4 (1/0)	8 (1/1)+
ADDQ	Mot Long	8 (1/0)	8 (1/0)	12 (1/2)+
ANDI	Octet,Mot		8 (2/0)	12 (2/1)+
ANDI	Mot Long		16 (3/0)	20 (3/2)+
CMPI	Octet,Mot		8 (2/0)	8 (2/0)+
CMPI	Mot Long		14 (3/0)	12 (3/0)+
EORI	Octet,Mot		8 (2/0)	12 (2/1)+
EORI	Mot Long		16 (3/0)	20 (3/2)+
MOVEQ	Mot Long		4 (1/0)	
ORI	Octet,Mot		8 (2/0)	12 (2/1)+
ORI	Mot Long		16 (3/0)	20 (3/2)+
SUBI	Octet,Mot		8 (2/0)	12 (2/1)+
SUBI	Mot Long		16 (3/0)	20 (3/2)+
SUBQ	Octet,Mot	8 (1/0)*	4 (1/0)	8 (1/1)+
SUBQ	Mot Long	8 (1/0)	8 (1/0)	12 (1/2)+

NOTES: + Ajouter le temps d'obtention de l'adresse * Mot seulement

VI - 3 - 3 - 2 - 5 - Exécution des instructions à un seul opérande

Instruction	Taille	Registre	Mémoire
CLR	Octet, Mot	4 (1/0)	8 (1/1)+
CLR	Mot Long	6 (1/0)	12 (1/2)+
NBCD	Octet	6 (1/0)	8 (1/1)+
NEG	Octet, Mot	4 (1/0)	8 (1/1)+
NEG	Mot Long	6 (1/0)	12 (1/2)+
NEGX	Octet, Mot	4 (1/0)	8 (1/1)+
NEGX	Mot Long	6 (1/0)	12 (1/2)+
NOT	Octet, Mot	4 (1/0)	8 (1/1)+
NOT	Mot Long	6 (1/0)	12 (1/2)+
Scc	Octet, Cd vraie	4 (1/0)	8 (1/1)+
Scc	Octet, Cd fausse	6 (1/0)	8 (1/1)+
TAS	Octet	4 (1/0)	10 (1/1)+
TST	Octet, Mot	4 (1/0)	4 (1/0)+
TST	Mot Long	4 (1/0)	4 (1/0)+

NOTES:

+ Ajouter le temps d'obtention de l'opérande

VI - 3 - 3 - 2 - 6 - Exécution des instructions de décalage et de rotation

Instruction	Taille	Registre	Mémoire
ASL,ASR	Octet, Mot	6+2n (1/0)	8 (1/1)+
ASL,ASR	Mot Long	8+2n (1/0)	
LSL,LSR	Octet, Mot	6+2n (1/0)	8 (1/1)+
LSL,LSR	Mot Long	8+2n (1/0)	
ROL,ROR	Octet, Mot	6+2n (1/0)	8 (1/1)+
ROL,ROR	Mot Long	8+2n (1/0)	
ROXL,ROXR	Octet, Mot	6+2n (1/0)	8 (1/1)+
ROXL,ROXR	Mot Long	8+2n (1/0)	

NOTES:

+ Ajouter le temps d'obtention de l'opérande

n nombre de décalages ou de rotations

VI - 3 - 3 - 2 - 7 - Exécution des instructions de manipulation de bits

Instruction	Taille	N° en registre Dn		N° immédiat (#...)	
		Registre	Mémoire	Registre	Mémoire
BCHG	Octet		8 (1/1)+		12 (2/1)+
BCHG	Mot Long	8 (1/0)*		12 (2/0)*	
BCLR	Octet		8 (1/1)+		12 (2/1)+
BCLR	Mot Long	10 (1/0)*		14 (2/0)*	
BSET	Octet		8 (1/1)+		12 (2/1)+
BSET	Mot Long	8 (1/0)*		12 (2/0)*	
BTST	Octet		4 (1/0)+		8 (2/0)+
BTST	Mot Long	6 (1/0)		10 (2/0)	

NOTES:

+ Ajouter le temps d'obtention de l'adresse * Indique une valeur maximum

VI - 3 - 3 - 2 - 8 - Exécution des instructions conditionnelles

Instruction	Déplacement	Branch effectué	Branch non effectué
Bcc	Octet	10 (2/0)	8 (1/0)
Bcc	Mot	10 (2/0)	12 (2/0)
BRA	Octet	10 (2/0)	
BRA	Mot	10 (2/0)	
BSR	Octet	18 (2/2)	
BSR	Mot	18 (2/2)	
DBcc	Condition vraie		12 (2/0)
DBcc	Condition fausse	10 (2/0)	14 (3/0)

VI - 3 - 3 - 2 - 9 - Exécution des instructions multiprécision

Instruction	Taille	Opération de Dn,Dn	Opération de M,M
ADDX	Octet,Mot	4 (1/0)	18 (3/1)
ADDX	Mot Long	8 (1/0)	30 (5/2)
CMPM	Octet, Mot		12 (3/0)
CMPM	Mot Long		20 (5/0)
SUBX	Octet, Mot	4 (1/0)	18 (3/1)
SUBX	Mot Long	8 (1/0)	30 (5/2)
ABCD	Octet	6 (1/0)	18 (3/1)
SBCD	Octet	6 (1/0)	18 (3/1)

VI - 3 - 3 - 2 - 10 - Exécution des instructions de manipulation d'adresses

Instruction	(An)	d(An)	d(An,Rx)	xxx.W	xxx.L	d(PC)	d(pc,Rx)
JMP	8 (2/0)	10 (2/0)	14 (3/0)	10 (2/0)	12 (3/0)	10 (2/0)	14 (3/0)
JSR	16 (2/2)	18 (2/2)	22 (2/2)	18 (2/2)	20 (3/2)	18 (2/2)	22 (2/2)
LEA	4 (1/0)	8 (2/0)	12 (2/0)	8 (2/0)	12 (3/0)	8 (2/0)	12 (2/0)
PEA	12 (1/2)	16 (2/2)	20 (2/2)	16 (2/2)	20 (3/2)	16 (2/2)	20 (2/2)

Instruction	(An)	(An)+	d(An)	d(An,Rx)	xxx.W	xxx.L	d(PC)	d(pc,Rx)
MOVEM								
M → R	12+ 3/0+	12+ 3/0+	16+ 4/0+	18+ 4/0+	16+ 4/0+	20+ 5/0+	16+ 4/0+	18+ 4/0+

NOTE:

Il convient d'ajouter:

- au nombre de périodes d'horloge pour des mots 4n avec n nombre de registre pour des mots longs 8n
- au nombre de lectures pour des mots n, pour des mots longs 2n

Instruction	(An)	-(An)	d(An)	d(An,Rx)	xxx.W	xxx.L
MOVEM						
R → M	8+ 2/n+	8+ 2/n+	12+ 3/n+	14+ 3/n+	12+ 3/n+	16+ 4/n+

NOTE:

- Il convient d'ajouter au nombre de périodes d'horloge pour des mots 4n avec n nombre de registre pour des mots longs 8n
- Le nombre d'écritures est égal à n pour les mots et 2n pour des mots longs

Instruction	Taille	R Æ M	M → R
MOVEP	Mot	16 (2/2)	16 (4/0)
MOVEP	Mot Long	24 (2/4)	24 (6/0)

VI - 3 - 3 - 2 - 11 - Exécution des instructions diverses

Instruction	Registre		Mémoire	
ANDI to (CCR ou SR)	20	(3/0)		
CHK	10	(1/0)+		
EORI to (CCR ou SR)	20	(3/0)		
ORI to (CCR ou SR)	20	(3/0)		
MOVE from SR	6	(1/0)	8	(1/1)+
MOVE to (CCR ou SR)	12	(2/0)	12	(2/0)+
EXG	6	(1/0)		
EXT (mot)	4	(1/0)		
EXT (mot long)	4	(1/0)		
LINK	16	(2/2)		
MOVE (to ou from) USP	4	(1/0)		
NOP	4	(1/0)		
RESET	132	(1/0)		
RTE	20	(5/0)		
RTR	20	(5/0)		
RTS	16	(4/0)		
STOP	4	(0/0)		
SWAP	4	(1/0)		
TRAPV (non accept)	4	(1/0)		
UNLK	12	(3/0)		

NOTE:+ Ajouter le temps d'obtention de l'adresse

VI - 3 - 3 - 3 - temps d'exécution des exceptions

Exception	Périodes	(Lct/Ect)
Erreur Adresse	50	(4/7)
Erreur Bus	50	(4/7)
CHK (excep effectuée)	44	(5/4)+
Division par 0	42	(5/4)
Instruction Illégale	34	(4/3)
Interruption	44	(5/3)*
Viol de privilège	34	(4/3)
Reset**	40	(6/0)
Trace	34	(4/3)
Instruction trappe	38	(4/4)
TrapV (excep effectuée)	34	(4/3)

NOTES:

+ Ajouter le temps d'obtention de l'adresse

* Le cycle de reconnaissance d'interruption est supposé prendre 4 cycles d'horloge

** Indique le temps qui sépare la désactivation de reset et halt du début de la 1ère instruction

VII - PROCESSUS D'ÉCHANGE, CHRONOGRAMMES

VII - 1 - Echanges Asynchrones

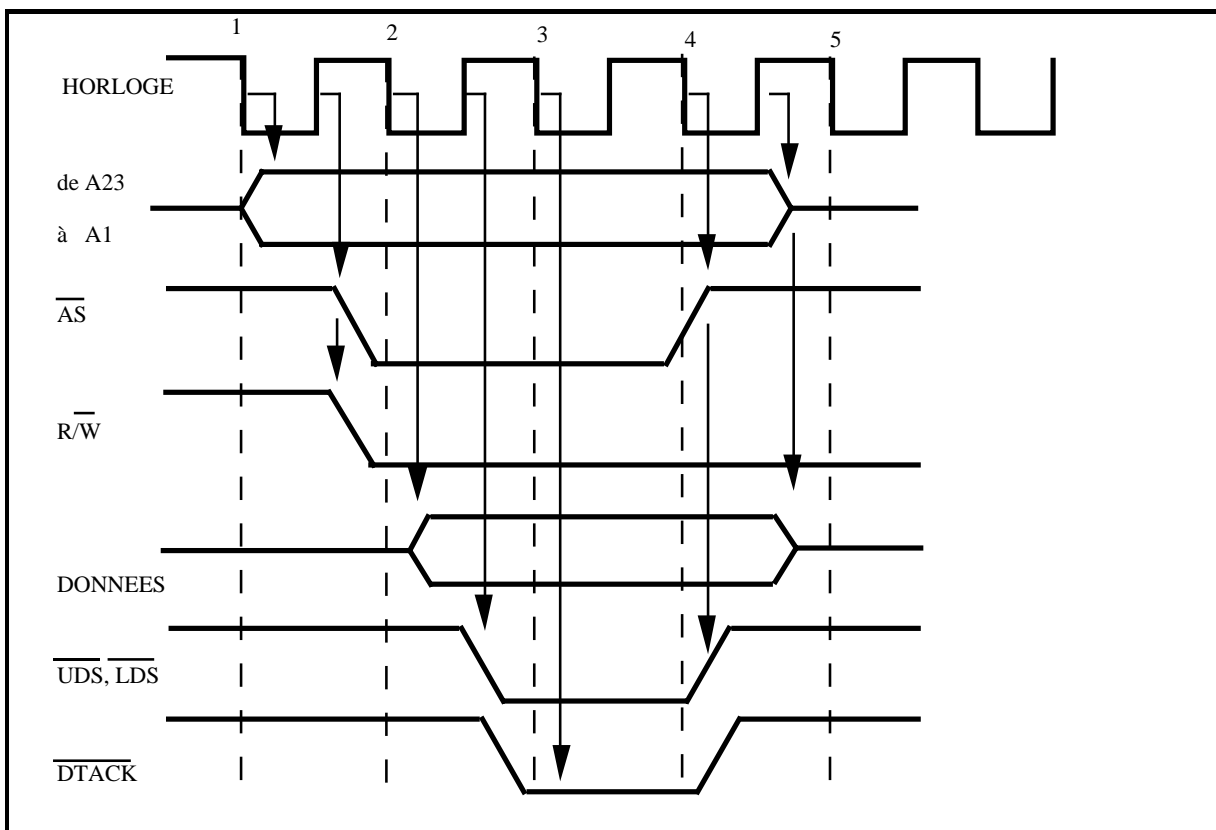
Les procédures d'échange asynchrones permettent la communication entre deux systèmes munis chacun d'une horloge dont les fréquences peuvent être très différentes.

Le mode d'échange asynchrone est basé sur la procédure dite de "Handshake" ainsi le cycle d'écriture se décompose comme suit:

- Le 68000 place adresse et donnée sur les bus et valide son adresse
- Le circuit de décodage informe le périphérique destinataire
- Le périphérique verrouille la donnée et accuse réception au 68000
- Le 68000 passe à l'instruction suivante.

VII - 1 - 1 - Cycle d'écriture asynchrone

Le chronogramme ci-dessous représente un cycle d'écriture asynchrone simplifié. Si le périphérique est suffisamment rapide pour mémoriser la donnée immédiatement et ne pas entraîner de cycles d'attente, une écriture s'effectue en 4 cycles d'horloge. Si le microprocesseur est piloté par une horloge à 16 MHz ceci représente 250 ns.



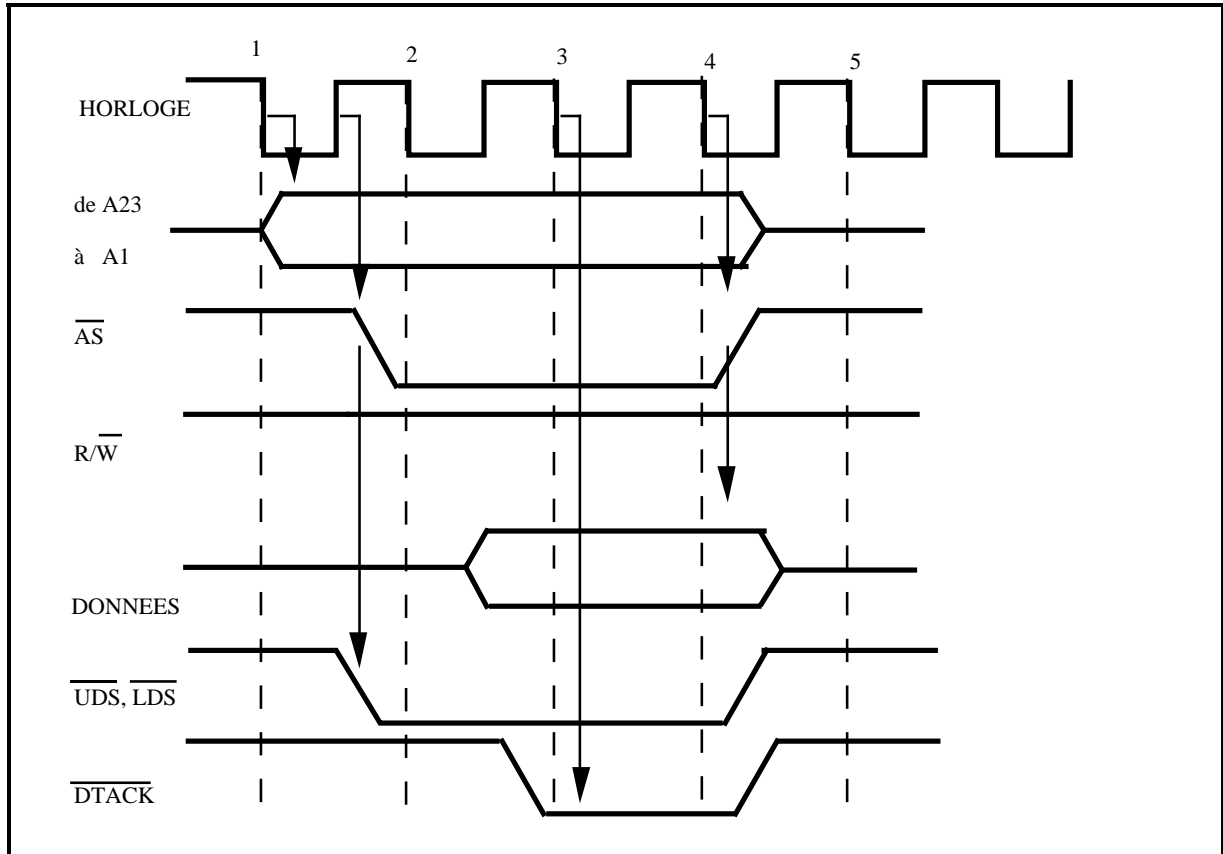
- 1/ Sur le premier front de descente d'horloge, le microprocesseur positionne ses bits d'adresse
- 2/ Sur le front de montée suivant il positionne son R/W à 0 et valide son adresse
- 3/ Sur le front de descente du second cycle les données sont placées sur le bus
- 4/ Sur le front de montée suivant le LDS et/ou UDS sont positionnés
- 5/ Si le périphérique est suffisamment rapide il retourne vers le 68000 le LDS ou UDS sur le DTACK qui sera pris en compte sur le front de descente du troisième cycle.
- 6/ Sur le front de descente du quatrième cycle le 68000 désactive AS et LDS (UDS) et le périphérique peut relâcher le DTACK
- 7/ La dernière étape se déroule sur le front de montée du quatrième cycle, le 68000 désactive ses

bus

VII - 1 - 2 - Cycle de lecture asynchrone

Le cycle de lecture comme le cycle d'écriture nécessite quatre périodes d'horloge si le périphérique est suffisamment rapide pour ne pas provoquer des cycles d'attente.

La figure ci-dessous représente un cycle de lecture asynchrone sans cycle d'attente

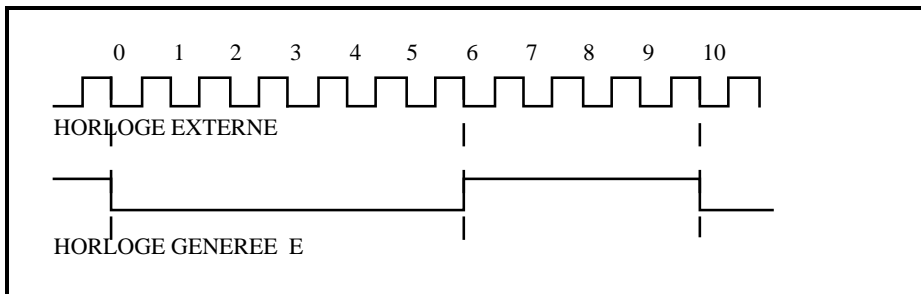


- 1/ Sur le premier front de descente d'horloge, le microprocesseur positionne ses bits d'adresse
- 2/ Sur le front de montée suivant il positionne son UDS et LDS à 0 et valide son adresse
- 3/ Le second cycle laisse au périphérique le temps d'extraire la donnée recherchée (temps d'accès)
- 4/ Sur le front de descente du troisième cycle, si le périphérique a tiré le DTACK à 0, le 68000 est informé de la présence des données sur le bus. Si non il vérifiera l'état du DTACK sur chaque front de descente jusqu'à ce qu'il enregistre un 0.
- 5/ Sur le front de descente du quatrième cycle le 68000 mémorise la donnée et désactive AS et LDS (UDS), le périphérique peut alors relâcher le DTACK
- 6/ Sur le front de montée du quatrième cycle, le 68000 désactive ses adresses.

VII - 2 - Echanges Synchrones

La technologie des périphériques de la famille 6800 n'admet pas des fréquences d'horloge aussi élevées et nécessitent un signal de synchronisation pour mémoriser la donnée. La fréquence de l'horloge du 68000 qui est actuellement de 8 à 16 MHz est incompatible avec celle des périphériques de la famille 6800 qui est de l'ordre de 1MHz. Afin de permettre des échanges synchrones avec les circuits de la famille 6800, le 68000 génère un signal d'horloge E en divisant par 10 la fréquence du signal CLK fourni au microprocesseur.

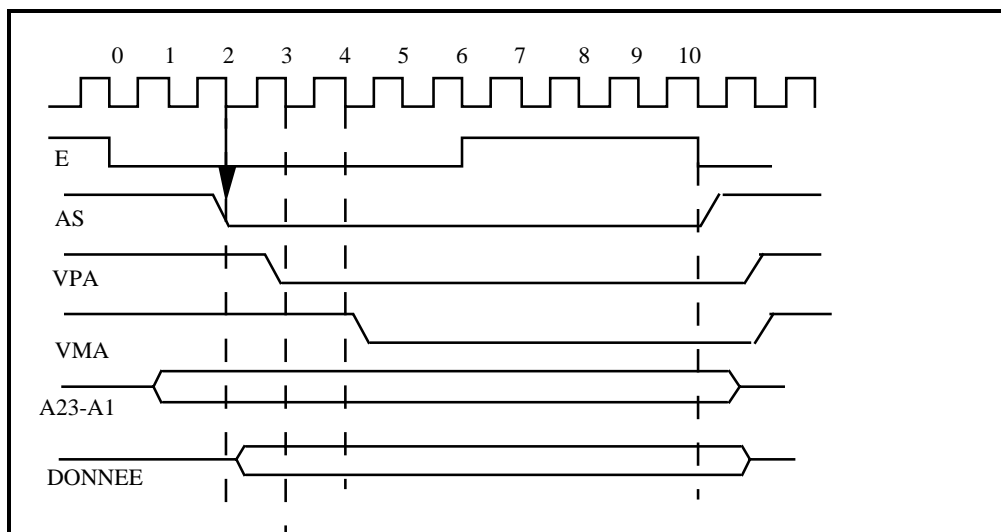
Le signal E généré par le 68000 reste à 0 pendant 6 périodes de l'horloge Clk et à 1 pendant 4, voir le croquis ci-dessous:



L'écoulement des instructions n'étant pas synchronisé sur ce signal E, il s'en suit que le microprocesseur peut faire une opération d'écriture ou de lecture à un moment quelconque par rapport à ce signal. Il est nécessaire de laisser au périphérique un temps suffisant pour décoder l'adresse et accéder à l'emplacement destination ou source. La donnée devant être mémorisée par le périphérique sur le front de descente du signal E, si cette opération s'effectue à un moment trop proche du front de descente de E, celui-ci n'aura pas le temps de prendre en compte la donnée.

Pour éviter un fonctionnement aléatoire, le circuit de décodage qui doit agir sur le CS du périphérique retourne ce signal sur la broche VPA qui sera détecté par le 68000 sur le premier front de l'horloge Clk. Si celui-ci est à 3 cycles d'horloge ou plus du front de montée de E, le 68000 active le signal VMA une période d'horloge après et le cycle s'achèvera sur le front de descente de E. Si le VPA est plus proche du front de montée de E, le VMA ne sera activé que durant le cycle suivant. Le cas le plus favorable (Best Case) représenté ci-dessous l'écriture se fait en 9 cycles d'horloge après l'activation du signal AS, le pire (Worst Case) nécessitera 18 cycles.

Cycle d'écriture synchrone, cas le plus favorable:



- 1/ - Sur le premier front de descente d'horloge après le passage à 0 de E le 68000 positionne ses adresses
- 2/ - Sur le front de descente suivant le 68000 positionne son Address Strobe
- 3/ - Le circuit de décodage du périphérique synchrone concerné retourne le signal VPA qui peut être échantillonné par le microprocesseur sur le front de descente du troisième cycle d'horloge.
- 4/ - Le front de descente du 4ème cycle d'horloge voit le 68000 activer son VMA validant ainsi le décodage d'adresse et le CS du périphérique soit un cycle avant la remontée de E et 5 avant sa redescente
- 5/ - Sur le front de descente de E, le 68000 désactive AS
- 6/ - Sur le front de descente suivant c'est le tour des données et des adresses d'être désactivés.

VII - 3 - Partage de la maîtrise des bus

Le 68000 possède trois lignes destinées à l'arbitrage de la maîtrise des bus:

$\overline{\text{BR}}$	-	Bus Request - Demande de la maîtrise des BUS
$\overline{\text{BG}}$	-	Bus Grant - BUS accordés
$\overline{\text{BGACK}}$	-	Bus Acknowledge - Accusé de prise de possession des BUS

VII - 3 - 1 - Principe de l'arbitrage

Le circuit demandant la maîtrise des bus tire la ligne $\overline{\text{BR}}$ à 0, le 68000 enregistre l'activation de ce signal sur le premier front descendant de son horloge et, suivant le moment où cet événement survient, le microprocesseur active sa sortie $\overline{\text{BG}}$ et met ses bus en haute impédance (1,5 à 4 périodes après la reconnaissance de BR). Le circuit demandeur peut alors prendre la maîtrise des bus, il le signale en tirant la ligne $\overline{\text{BGACK}}$ à 0.

Lorsque le circuit qui a demandé la maîtrise des bus passe à une autre activité, il relâche le DTACK. Le 68000 enregistre cette remontée sur son premier front de descente d'horloge. Une période et demi d'horloge lui est nécessaire pour reprendre son activité interrompue.

VII - 3 - 2 - Chronogrammes d'arbitrage des bus

Les trois chronogrammes qui suivent, extraits de la documentation Motorola font apparaître trois scénarios différents suivant le moment où survient la demande

- Bus inactifs, cas le plus favorable
- Bus actifs
- Les bus sont sur le point d'être activés, une décision interne au microprocesseur vient d'être prise, cas le plus défavorable.

Lorsque plusieurs circuits peuvent prétendre à la maîtrise des bus, leurs lignes BR peuvent être câblées en OU logique (logique négative). Dans le cas où plusieurs circuits demanderaient l'accès aux bus simultanément, un circuit permettant d'établir des priorités doit être utilisé.

Le signal $\overline{\text{BGACK}}$ ne doit être activé que lorsque les lignes $\overline{\text{AS}}$, $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, FC ainsi que les bus sont en haute impédance. Le demandeur est maître des bus tant que sa ligne $\overline{\text{BGACK}}$ est à 0. L'accès aux bus se termine lorsque la ligne $\overline{\text{BGACK}}$ est relâchée (à 1).

Lorsqu'un circuit accède à la maîtrise des bus, $\overline{\text{BGACK}} = 0$, il doit relâcher sa ligne $\overline{\text{BR}}$ mais si plusieurs circuits sont demandeurs, la ligne étant câblée en OU logique, celle-ci restera à 0. Dès l'accès terminé, $\overline{\text{BGACK}} = 1$, si $\overline{\text{BR}}$ est toujours à 0, le MPU reprend aussitôt un cycle d'attribution des bus.

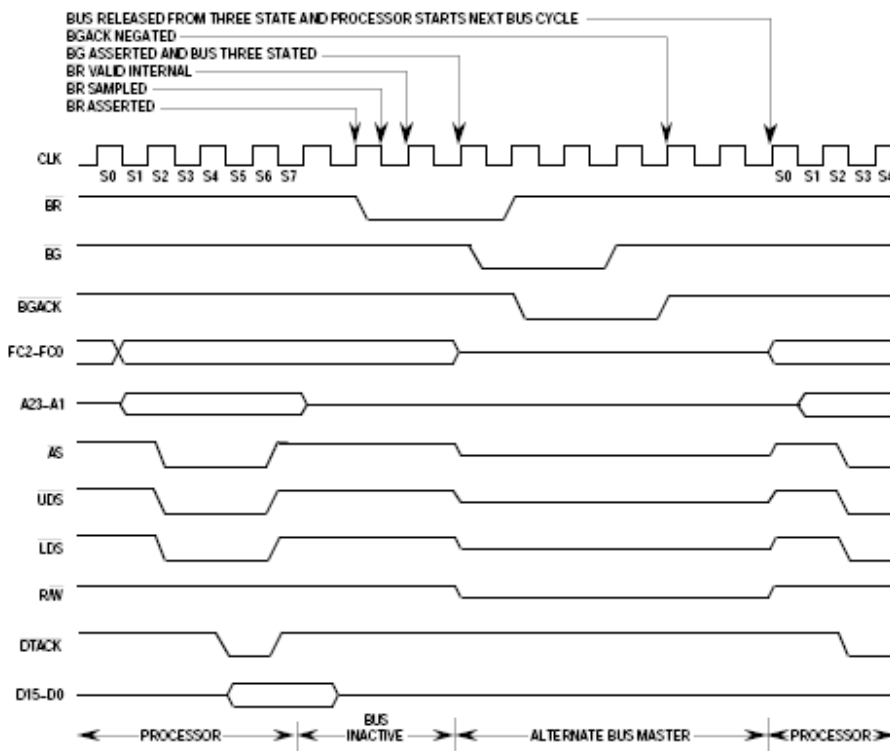


Figure 5-20. 3-Wire Bus Arbitration Timing Diagram—Bus Inactive

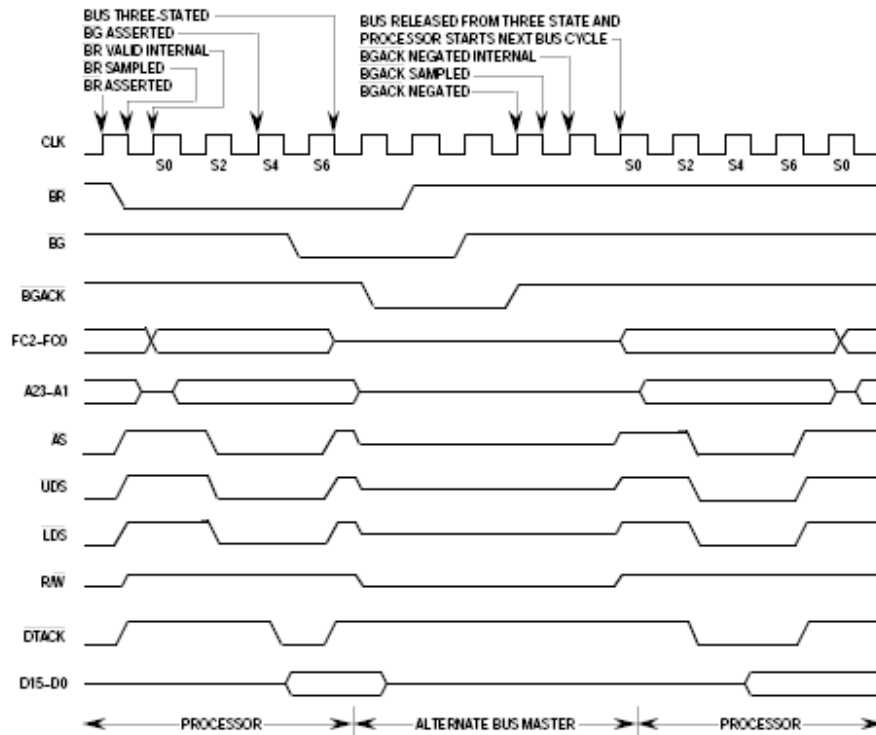


Figure 5-21. 3-Wire Bus Arbitration Timing Diagram—Special Case

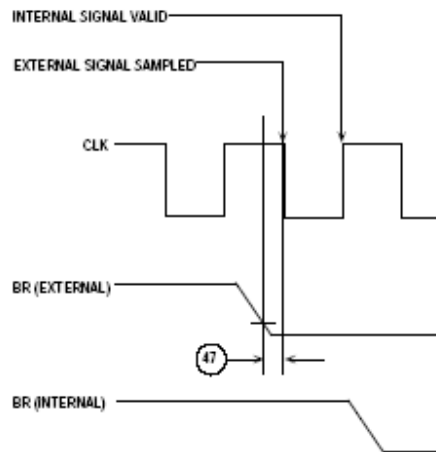


Figure 5-17. External Asynchronous Signal Synchronization

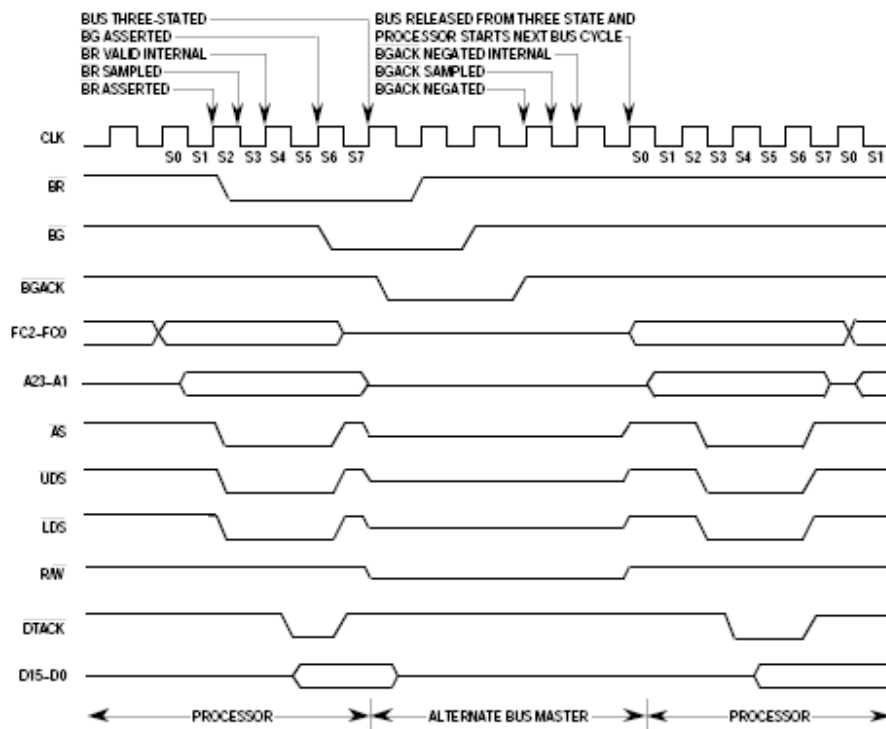
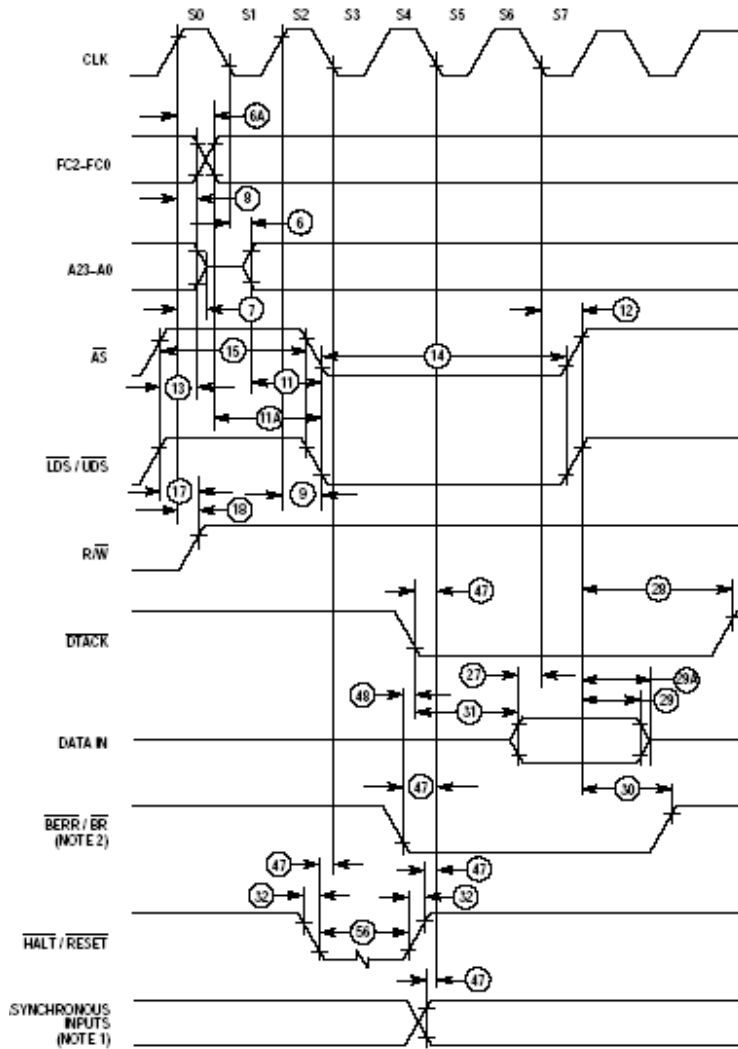


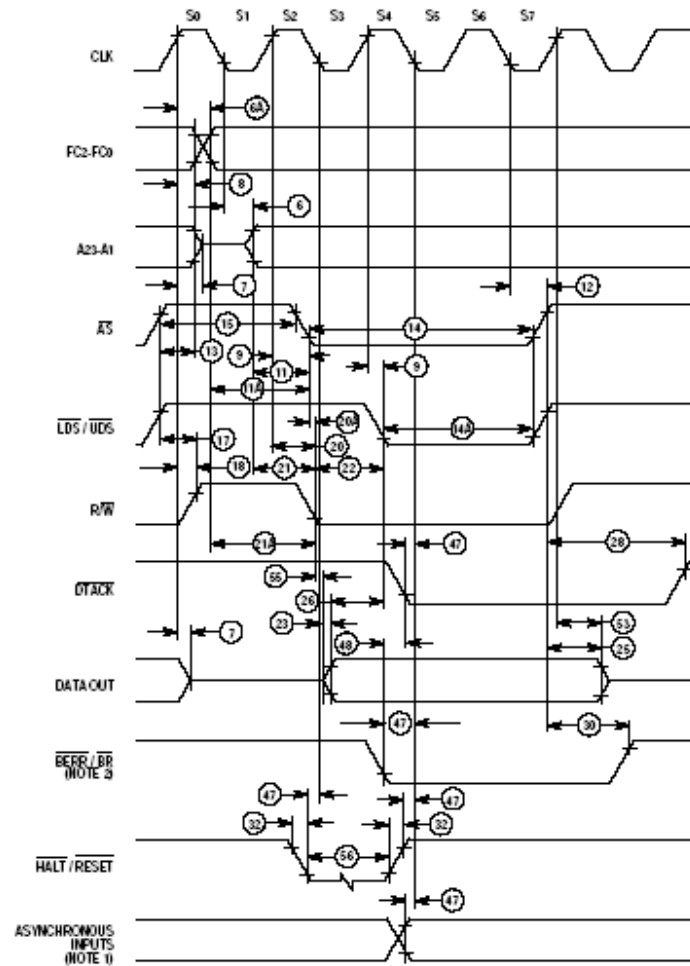
Figure 5-19. 3-Wire Bus Arbitration Timing Diagram—Processor Active

VIII - SPECIFICATIONS DU CONSTRUCTEUR

VIII - 1 - Cycle de lecture en mode Asynchrone



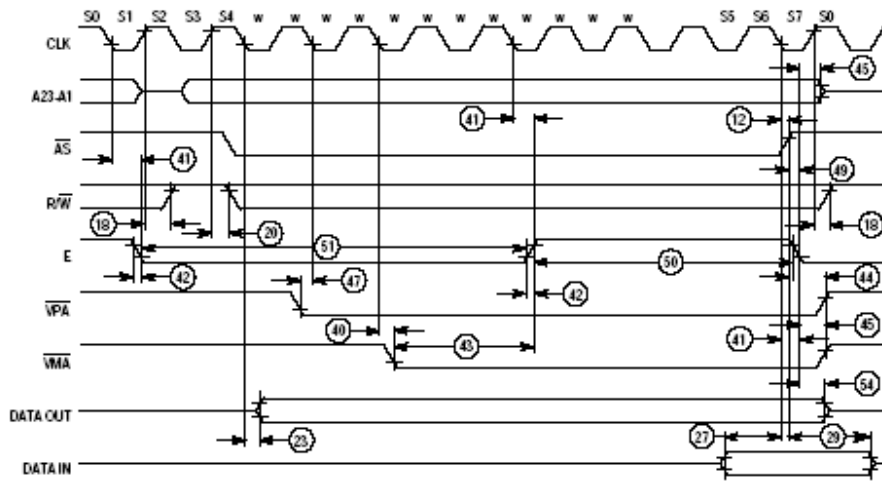
VIII - 2 - Cycle d'écriture en mode Asynchrone



- NOTES:
1. Timing measurements are referenced to and from a low voltage of 0.6 V and a high voltage of 2.0 V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between 0.6 V and 2.0 V.
 2. Because of loading variations, R/W may be valid after AS even though both are initiated by the rising edge of S2 (specification #20A).

Figure 10-5. Write Cycle Timing Diagram
(Applies To All Processors Except The MC88EC000)

VIII - 3 - Transfert en mode Synchronre



NOTE: This timing diagram is included for those who wish to design their own circuit to generate \overline{VMA} . It shows the best case possible attainable

Figure 10-6. MC68000 to M6800 Peripheral Timing Diagram (Best Case)
 (Applies To All Processors Except The MC88EC000)

10.11 AC ELECTRICAL SPECIFICATIONS—MC68000 TO M6800

PERIPHERAL ($V_{CC} = 5.0 \text{ Vdc} \pm 5\%$; $GND=0 \text{ Vdc}$; $T_A = T_L \text{ TO } T_H$; refer to figures 10-8)
(Applies To All Processors Except The MC68EC000)

Num	Characteristic	8 MHz*		10 MHz*		12.5 MHz*		16.67 MHz *12F*		16 MHz		20 MHz*		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
12 ¹	Clock Low to \overline{AS} , \overline{DS} Negated	—	62	—	50	—	40	—	40	3	30	3	25	ns
18 ¹	Clock High to R/ \overline{W} High (Read)	0	55	0	45	0	40	0	40	0	30	0	25	ns
20 ¹	Clock High to R/ \overline{W} Low (Write)	0	55	0	45	0	40	0	40	0	30	0	25	ns
23	Clock Low to Data-Out Valid (Write)	—	62	—	50	—	50	—	50	—	30	—	25	ns
27	Data-In Valid to Clock Low (Setup Time on Read)	10	—	10	—	10	—	7	—	5	—	5	—	ns
29	\overline{AS} , \overline{DS} Negated to Data-In Invalid (Hold Time on Read)	0	—	0	—	0	—	0	—	0	—	0	—	ns
40	Clock Low to \overline{VMA} Asserted	—	70	—	70	—	70	—	50	—	50	—	40	ns
41	Clock Low to E Transition	—	55	—	45	—	35	—	35	—	35	—	30	ns
42	E Output Rise and Fall Time	—	15	—	15	—	15	—	15	—	15	—	12	ns
43	\overline{VMA} Asserted to E High	200	—	150	—	90	—	80	—	80	—	60	—	ns
44	\overline{AS} , \overline{DS} Negated to \overline{VPA} Negated	0	120	0	90	0	70	0	50	0	50	0	42	ns
45	E Low to Control, Address Bus Invalid (Address Hold Time)	30	—	10	—	10	—	10	—	10	—	10	—	ns
47	Asynchronous Input Setup Time	10	—	10	—	10	—	10	—	10	—	5	—	ns
49 ²	\overline{AS} , \overline{DS} , Negated to E Low	-70	70	-55	55	-45	45	-35	35	-35	35	-30	30	ns
50	E Width High	450	—	350	—	280	—	220	—	220	—	190	—	ns
51	E Width Low	700	—	550	—	440	—	340	—	340	—	290	—	ns
54	E Low to Data-Out Invalid	30	—	20	—	15	—	10	—	10	—	5	—	ns

*These specifications represent improvement over previously published specifications for the 8-, 10-, and 12.5-MHz MC68000 and are valid only for product bearing date codes of 8827 and later.

** This frequency applies only to MC68HC000 and MC68HC001.

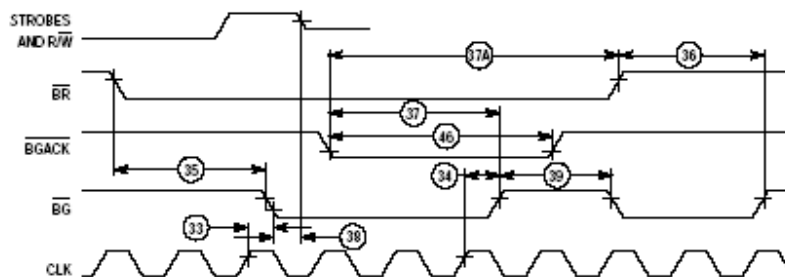
NOTES: 1. For a loading capacitance of less than or equal to 50 pF, subtract 5 ns from the value given in the maximum columns.
2. The falling edge of S_6 triggers both the negation of the strobes (\overline{AS} and \overline{DS}) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specification #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.

VIII - 4 - Arbitrage des bus

10.12 AC ELECTRICAL SPECIFICATIONS — BUS ARBITRATION ($V_{CC}=5.0$

$V_{DC}\pm 5\%$; $GND=0$ VDC, $T_A=T_L$ TO T_H ; See Figures 10-7 – 10-11) (Applies To All Processors Except The MC88EC000)

Num	Characteristic	8 MHz*		10 MHz*		12.5 MHz*		16.67 MHz 12F		16 MHz		20 MHz*		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
7	Clock High to Address, Data Bus High Impedance (Maximum)	—	80	—	70	—	60	—	50	—	50	—	42	ns
16	Clock High to Control Bus High Impedance	—	80	—	70	—	60	—	50	—	50	—	42	ns
33	Clock High to \overline{BG} Asserted	—	62	—	50	—	40	0	40	0	30	0	25	ns
34	Clock High to \overline{BG} Negated	—	62	—	50	—	40	0	40	0	30	0	25	ns
35	\overline{BR} Asserted to \overline{BG} Asserted	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Ckls
36 ¹	\overline{BR} Negated to \overline{BG} Negated	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Ckls
37	\overline{BGACK} Asserted to \overline{BG} Negated	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Ckls
37A ²	\overline{BGACK} Asserted to \overline{BR} Negated	20	1.5 Ckls	20	1.5 Ckls	20	1.5 Ckls	10	1.5 Ckls	10	1.5 Ckls	10	1.5 Ckls	Ckls/ns
38	\overline{BG} Asserted to Control, Address, Data Bus High Impedance (\overline{AS} Negated)		80		70		60	—	50	—	50	—	42	ns
39	\overline{BG} Width Negated	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Ckls
46	\overline{BGACK} Width Low	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Ckls
47	Asynchronous Input Setup Time	10	—	10	—	10	—	5	—	5	—	5	—	ns
57	\overline{BGACK} Negated to \overline{AS} , \overline{DS} , R/W Driven	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Ckls
57A	\overline{BGACK} Negated to FC, \overline{VMA} Driven	1	—	1	—	1	—	1	—	1	—	1	—	Ckls
58 ¹	\overline{BR} Negated to \overline{AS} , \overline{DS} , R/W Driven	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	Ckls
58A ¹	\overline{BR} Negated to FC, \overline{VMA} Driven	1	—	1	—	1	—	1	—	1	—	1	—	Ckls



NOTE: Setup time to the clock (#47) for the asynchronous inputs \overline{BERR} , \overline{BGACK} , \overline{BR} , \overline{DTACK} , $\overline{IPL2}$ - $\overline{IPL0}$, and \overline{VPA} guarantees their recognition at the next falling edge of the clock.

Figure 10-7. Bus Arbitration Timing
(Applies To All Processors Except The MC88EC000)

IX - LES BOITIERS

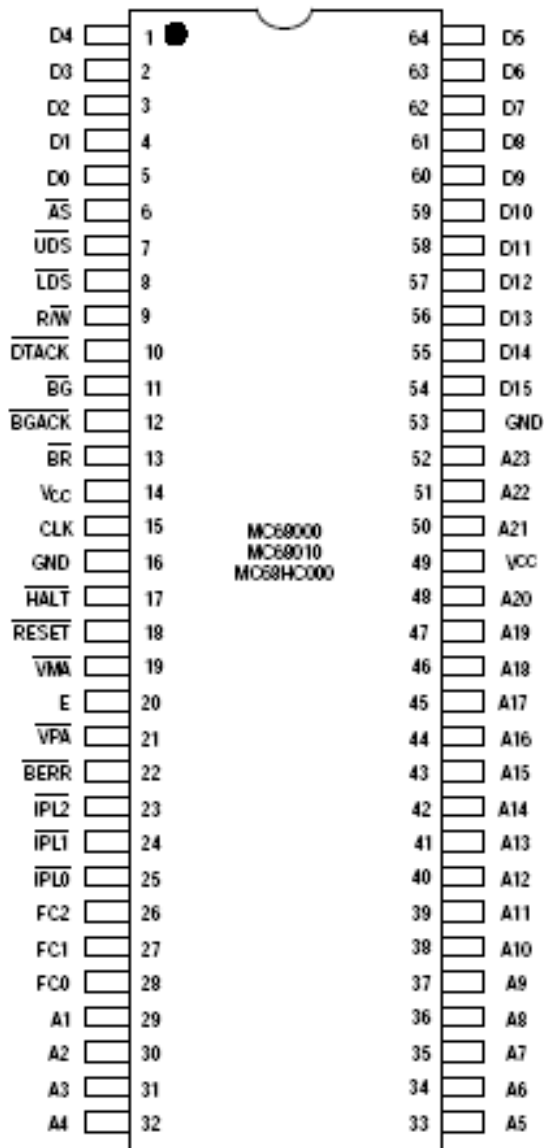


Figure 11-1. 64-Pin Dual In Line

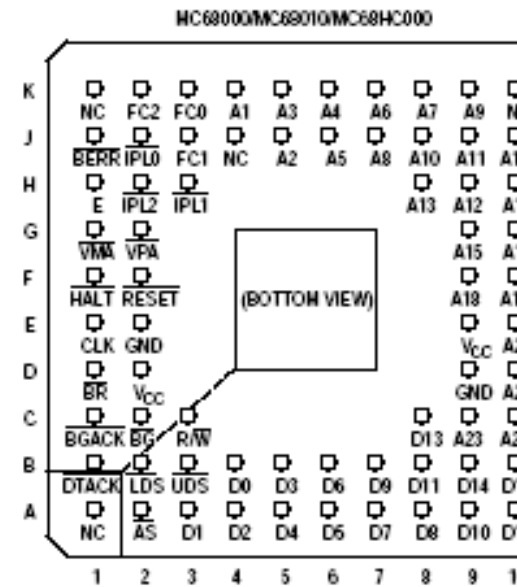


Figure 11-2. 68-Lead Pin Grid Array

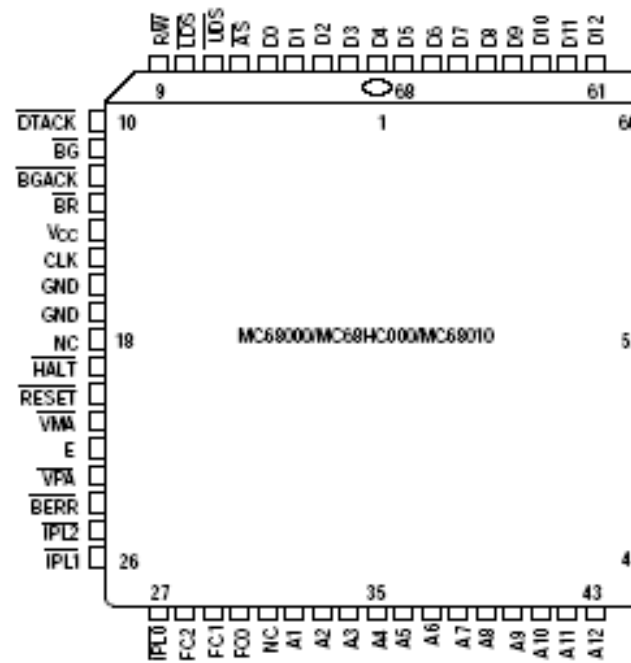


Figure 11-3. 68-Lead Quad Pack