



5.9- Les mécanismes d'exceptions logicielles

Les microprocesseurs et les microcontrôleurs intègrent des mécanismes propres à donner des réponses satisfaisantes à, soit à des événements logiciels « anormaux », il s'agit des exceptions, soit à des événements physiques issus de capteurs câblés sur des entrées particulières, il s'agit des interruptions.

Les microprocesseurs offrent les deux mécanismes, gestions d'exceptions et d'interruptions. En général les microcontrôleurs n'offrent que les mécanismes liés aux gestions d'interruptions.

Les exceptions logicielles sont des comportements « réflexes » adoptés par le microprocesseur à l'occasion de l'exécution d'une instruction « anormale ».

Typiquement il est possible de classer ces exécutions en 4 grandes catégories.

Exécution d'instructions « normales » mais qui posent problème à l'instant de l'exécution.

Exemple, une division (divs d1,d0), sera exécutée des millions de fois sans « problème » jusqu'au moment où le diviseur (d1) adoptera (il contient, exemple, une valeur issue d'un CAN) la valeur zéro. Alors le microprocesseur n'exécutera pas l'instruction et se déroutera vers l'exécution d'un programme particulier (à écrire) censé répondre au problème rencontré d'une division par zéro. Ce type d'exception est noté 1 dans la table d'exceptions page : 128

Exécution d'instructions « privilégiées » exécutées dans un contexte non privilégié.

Le Coldfire, dispose de deux modes de fonctionnement. Le mode privilégié dit « superviseur » et le mode non privilégié dit « user ». Si une instruction dite privilégiée, telle par exemple que `move.w d0,sr` est exécutée en mode non privilégié (user) alors, le microprocesseur n'exécutera pas l'instruction et se déroutera vers l'exécution d'un programme particulier (à écrire) censé répondre au problème rencontré d'une violation de privilèges. Par « définition » un tel scénario ne peut pas se présenter lors de l'exécution d'un programme qui se déroulerait en mode superviseur. Ce type d'exception est noté 2 dans la table d'exceptions page : 128

Exécutions d'instructions « anormales ».

Le Coldfire a un jeu d'instructions codé sur 16 bits. Ce code lui confère un potentiel de 2^{16} (65536) instructions distinctes. Le jeu d'instructions effectif n'offre pas autant d'instructions. Il existe donc des codes sans matérialisation d'instruction associée. Si par hasard, le processeur « pointe », sur un tel code, plutôt que d'exécuter « n'importe quoi », il se déroutera vers l'exécution d'un programme particulier (à écrire) censé répondre au problème rencontré. Ce type d'exception est noté 3 dans la table d'exceptions page : 128

Exécutions d'instructions « normales », dites exceptions logicielles (TRAP).

Il est possible, quel que soit le mode de fonctionnement du microprocesseur, de forcer un mécanisme d'exception via l'usage de l'instruction TRAP #n ($n \in [0,15]$).

Alors l'exécution du programme qui contient l'instruction TRAP #n, se déroutera vers l'exécution d'un programme particulier parmi 15. Ce type d'exception est noté 4 dans la table d'exceptions page : 128



Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

Tous ces traitements d'exception sont caractérisés par le séquençement suivant :

-Sauvegarde interne (dans le microprocesseur) du registre d'état SR tel qu'il est au moment de l'exécution de l'exception. Cela signifie en particulier, que le mode courant (user ou superviseur, matérialisé par l'état du bit S de SR) est sauvé.

-modification des bits T et S mis respectivement à 0 et 1.

T ← 0, cela signifie que si le microprocesseur était en mode trace (pas à pas) au moment de l'exception, le programme de traitement de cette exception se déroulera lui à « pleine vitesse *».

S ← 1, cela signifie que « la suite » des exécutions (en fait jusqu'à l'exécution de l'instruction RTE à placer au bout du programme de traitement de l'exception) sera réalisée en mode superviseur (privilégié).

*cette expression, pleine vitesse, veut s'opposer à l'expression pas à pas.

-le microprocesseur détermine le numéro du vecteur qui représente le lien entre l'exception détectée et la table d'exceptions (page : 128). Par exemple, la division par zéro est attachée au vecteur numéro 5.

-le microprocesseur sauve sur la pile 3 éléments (1 long + deux mots).

°la valeur de PC courante (1long). (parfois l'adresse même de l'instruction qui est à l'origine de l'exception, c'est le cas par exemple de la division par zéro, d'autre fois l'adresse de l'instruction qui suit l'instruction qui est à l'origine de l'exception, c'est le cas par exemple de l'instruction TRAP #n) Ces deux scénarios sont précisés dans la table d'exceptions (page : 128 colonne : adresse de retour)

dans la rubrique « adresse de retour (suite exécution RTE) ». Pour le second cas, une instruction RTE placée en fin de programme d'exceptions, conduira « naturellement » à la poursuite du programme « principal ». Dans le premier cas, la même instruction RTE conduira à une répétition de l'instruction origine de l'exception dans le programme « principal ». En d'autre terme, pour ces derniers cas, il est nécessaire de régler par logiciel, dans le module de traitement de l'exception, la procédure de retour.

°un mot (16 bits) qui contient diverses informations, dont le numéro du vecteur associé à l'exception.

°un mot (16 bits) qui correspond à la valeur initiale du registre SR (en fait la valeur de SR sauvée en interne, dans la première phase de cette procédure).

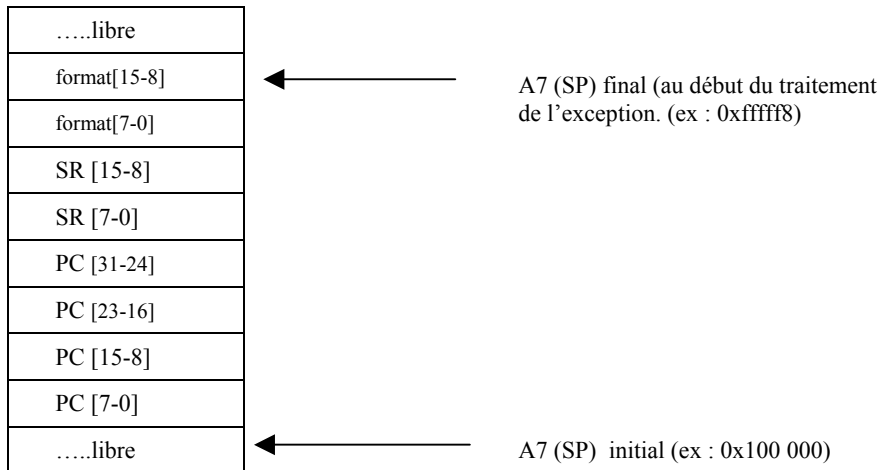
Remarque : ces valeurs sont sauvées sur la pile en utilisant une valeur de pointeur de pile (a7) toujours modulo 4 (valeur terminée par deux zéro). Cela optimise la vitesse de gestion de la pile pour la suite.



Le microprocesseur *Cold FIRE 5307*

C. GUIRAUDIE

Le contexte représenté sous forme d'une suite d'octets en mémoire vive (pile) est donc:



Le contexte sauvé est décrit précisément page : 124 .

-le compteur ordinal PC est chargé avec l'adresse rangée dans la table d'exception au point d'entrée fixé par le vecteur attaché à l'exception traitée.

PC= long d'adresse (VBR) + 4x vecteur.

registre VBR (Vecteur Base Register) voir page : **Erreur ! Signet non défini.**

Ainsi si VBR = 0 et s'il s'agit par exemple de l'exécution de l'instruction TRAP #2, alors c'est à l'adresse 0 + (4 x 34), soit l'adresse 0x88 , que le long qui représente l'adresse de début du module d'exception à exécuter est lu et chargé par le processeur dans son registre PC. Ainsi l'instruction exécutée suivante sera bien la première instruction du programme d'exception attaché à l'instruction TRAP #2.

Le Coldfire, comme la plupart des produits modernes, gère les points d'entrées des exceptions (adresse de début des modules d'exception) à travers une table d'exception. Cette dernière est pointée par le registre VBR, elle contient 256 points d'entrées (256 adresses == 256 longs de 32 bits → taille de la table = 1024 octets)

Les détails de cette table sont donnés page131

Remarque: pour revenir d'un module de traitement d'exception vers le module « principal », il faut utiliser l'instruction RTE qui est la seule à même de restituer le contexte de pile décrit ci-avant.

Exemple de mise ne œuvre d'un programme nommé calcul, mise en œuvre via l'usage de l'instruction TRAP #15. (hypothèse VBR = 0 → début table en 0)

En assembleur :

```
....
    lea    calcul,d0
    move.l d0,0xbc      ;0xbc= 0x80 (début points d'entrées trap) + (15 x 4)
...
    trap  #15
....
```



Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

calcul :

```
....
rte                ;fin du module traitement d'exception
```

En C :

```
__interrupt__ void calcul(void)           // le mot clé __interrupt__ forcera
                                           // le compilateur à générer un RTE en lieu et
                                           // place de l'accolade fermante.
{
    ...                                    // corps de la fonction calcul
}
```

main()

```
{
int    *ptr ;
ptr = (int) 0xbc ;                        //point d'entrée de la table d'exception
*ptr = (int) calcul ;                    // à l'@ 0xbc est rangé l'@ de début de la
                                           // fonction calcul
...
_asm(«\ttrap #15\n\r») ;                 //appel « assembleur » de l'exception trap #15
...
}
```

5.10 Les mécanismes d'exceptions matérielles

Les interruptions sont des mécanismes intégrés dans tous les microprocesseurs ou les microcontrôleurs qui permettent d'obtenir un déroutement de l'exécution d'un programme lors de l'occurrence d'un signal physique câblé sur une broche spéciale appelée broche d'entrée d'interruption.

Ainsi au lieu de tester l'occurrence d'un signal issu d'un capteur par scrutation (polling) il est préférable, SI CE CAPTEUR DELIVRE UN SIGNAL D'OCCURRENCE RARE (typiquement capteur d'incident) de le câbler sur l'une des entrées d'interruption du microprocesseur.

Il faut noter ici, que le nombre d'entrées d'interruption d'un microprocesseur est toujours très limité (de 1 à 8). En conséquence, le coût « d'attachement » d'une variable d'entrées sur une de ces broches est beaucoup plus coûteux que l'usage d'une broche de port d'entrées. Donc les entrées d'interruption sont typiquement à réserver aux seules entrées dites à occurrences (apparition) rares et non périodiques.

Avant de décider si un capteur sera géré en tant qu'interruption (ou traité par une logique câblée dédiée) il faut maîtriser les informations générales suivantes :

- une interruption n'est prise en compte par le microprocesseur, qu'à l'issue de l'exécution de l'instruction en cours.

- le temps de latence (temps qui s'écoule entre la prise en compte de l'interruption et l'exécution de la première instruction du module de traitement de l'interruption) est non nul (ex quelque micro secondes, à quelque dizaines de microsecondes).

- une interruption est plus ou moins prioritaire vis-à-vis d'une autre (si simultanéité d'occurrence).

En général, c'est un choix de câblage qui est associé à cette hiérarchie.



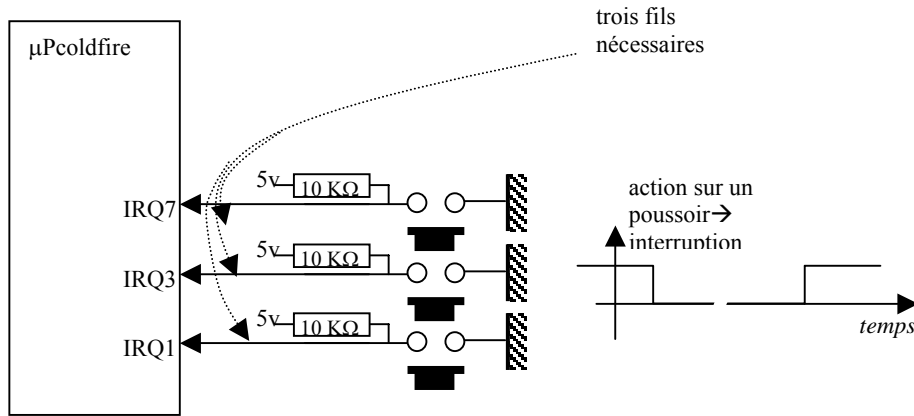
Le microprocesseur *Cold FIRE 5307*

C. GUIRAUDIE

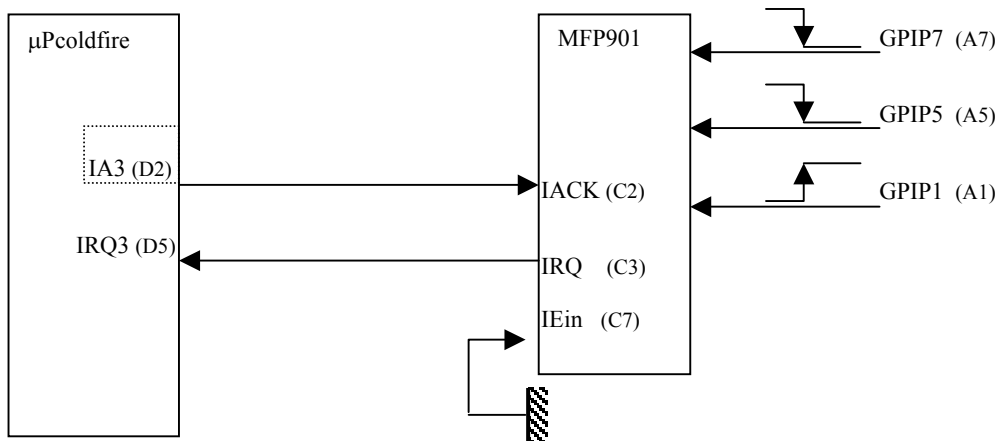
Usage des interruptions avec le microprocesseur COLDFIRE 5307.

Aspects matériel :

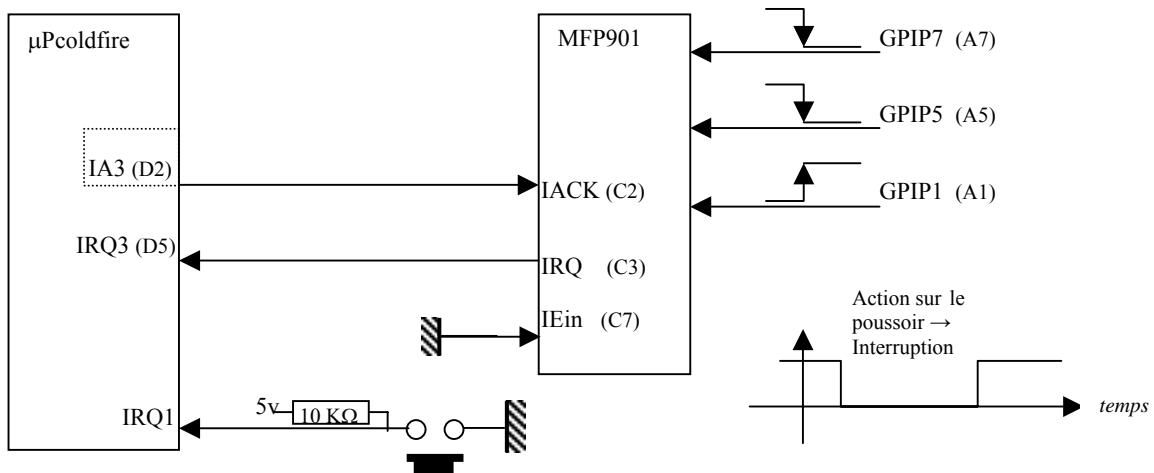
Exemple de possibilités de câblages en mode autovectorisé :



Exemple de possibilités de câblages en mode vectorisé :
(ici usage du contrôleur d'interruption MPF901)



Exemple de possibilités de câblages en mode vectorisé et autovectorisé :
(ici usage du contrôleur d'interruption MPF901)





Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

Les broches impliquées du microprocesseur:

4 broches d'entrées d'interruptions. → IRQ 7 ; IRQ 5 ; IRQ3 ; IRQ1

Attention la broche IRQ5 est utilisée par les ressources de la carte mère utilisée en TP, donc indisponible.

Sensibles sur niveau 0.

4 signaux d'acquiescement d'interruption. → IACK 7 ; IACK 3 ; IACK 3 ; IACK 1

Ces signaux sont à réaliser via une logique de décodage externe au microprocesseur.

Cette logique à base de circuits décodeurs (type 74128) fournie sur 7 broches les signaux IACK7 à 1 à partir des signaux présents sur les broches TT0_1 et TM 0-2 du microprocesseur collfire.

Hiérarchie de prise en compte des événements en cas d'arrivée simultanée :

IRQ7 > IRQ 5 > IRQ3 > IRQ1

Les sources d'interruptions attachées à ces broches sont « masquages » à plusieurs niveaux.

1° Au niveau du masque I2-I1-I0 du registre d'état du microprocesseur.

Remarques : Pour la carte mère utilisée en TP.

Suite à un reset, ces masques sont mis à 0-0-0 par le moniteur. De ce fait, tous les niveaux d'interruption sont admis :

Pour changer cela (niveau accepté seulement si > niveau du masque I2-I1-I0) :

-« à la main » tapez la commande : == > sr XYXX

le quarté Y valant de 0 à 7 suivant le masque désiré.

-par programme. Il est nécessaire d'être en mode superviseur (usage d'un TRAP par exemple) alors

→ move.w #0xXYXX,SR

2° Au niveau du registre de masquage interne IMR :

accès long , adresse MBAR + 0x44 sur la carte mère MBAR= 0xF000 0000

Topologie de ce registre :

31	«30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														DMA3	DMA2

RESET

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA1	DMA0	UART2	UART1	MBUS	TIMER2	TIMER1	SWT	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	

RESET

Registre IMR

Les sources IRQ 7 à 1 sont démasquées par l'écriture d'un 0 sur les bits correspondant EINT7 à EINT1 du registre.

ATTENTION POUR MANIPULER CE REGISTRE IL FAUT :

LECTURE + MASQUAGE (and pour mise à 0 ou or pour mise à 1) + ECRITURE.

Une écriture directe conduirait à « bloquer » des ressources de la carte mère.



Le microprocesseur *Cold FIRE 5307*

C. GUIRAUDIE

Exemples de programme :

Démasquer l'IT IRQ3, sans altérer les autres sources.

```
..... lea      0xf0000000+0x44,a1
      move.l   (a1),d0
      and.l    #0xffffffff7,d0
      move.l   d0,(a1)
.....
```

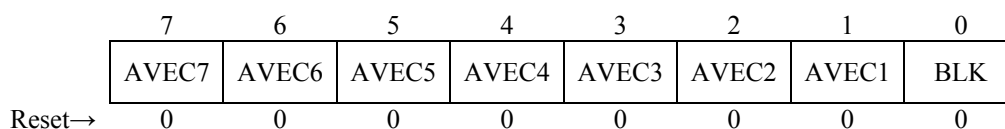
Masquer l'IT IRQ3, sans altérer les autres sources.

```
..... lea      0xf0000000+0x44,a1
      move.l   (a1),d0
      or.l     #0x00000008,d0
      move.l   d0,(a1)
.....
```

3° Au niveau du(es) registre(s) de masquage interne au contrôleur éventuellement utilisé :
Voir documentation spécifique (par exemple MFP 901)

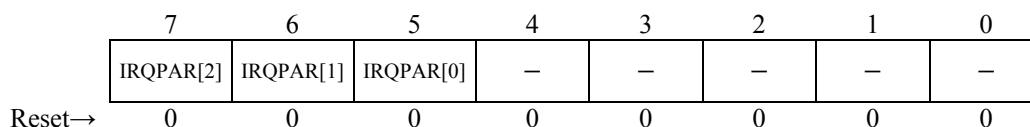
Le choix du mécanisme d'auto-vectorisation pour une source d'interruption donnée est possible via la mise 1 du bit adéquat du registre AVR (MBAR + 0x4B ; accès octet)

La topologie de ce registre est la suivante :



Les broches IRQ (IACK) 7-5-3-1 peuvent par programmation du registre IRQPAR (MBAR + 0x06 ; accès octet) « changer » de niveau et acquérir les niveaux « manquants » 2-4-6

La topologie de ce registre est la suivante :



Les bits IRQPAR ont les rôles suivants :

IRQPAR	IRQPAR = 0	IRQPAR = 1	EXTERNAL PIN CONNECTION
IRQPAR[2]	IRQ[5]	IRQ[4]	IRQ[5]
IRQPAR[1]	IRQ[3]	IRQ[6]	IRQ[3]
IRQPAR[0]	IRQ[1]	IRQ[2]	IRQ[1]

Aspects logiciels :

Le microprocesseur gère les mécanismes d'interruption et d'exception via une table dite d'exception.

Cette table doit contenir les adresses de début des modules (programmes) d'interruption à exécuter pour telle ou telle exception ou telle interruption.

Si le registre VBR du microprocesseur n'est pas modifié alors la table est débute à l'adresse 0. Elle fait 1024 octets. Elle peut contenir $1024/4 = 256$ adresses de début de programmes d'exception ou d'interruption.



Le microprocesseur *Cold FIRE 5307*

C. GUIRAUDIE

Le tableau ci-dessous donne le contenu de cette table d'exception :

Numéro Vecteur	Adresse Vecteur	Affectation	type	adresse de retour (suite exécution RTE)
00	0x000	Reset Initialisation de SSP		sans objet
01	0x004	Reset Initialisation de PC		sans objet
02	0x008	Erreur d'accès	1	instruction à l'origine de l'exception
03	0x00C	Erreur Adresse	1	instruction à l'origine de l'exception
04	0x010	Instruction Illégale	3	instruction à l'origine de l'exception
05	0x014	Division par zéro	1	instruction à l'origine de l'exception
06	0x018	réservée		sans objet
07	0x01C	réservée		sans objet
08	0x020	Viol de privilège	2	instruction à l'origine de l'exception
09	0x024	Trace		instruction suivante du programme
10	0x028	Emulateur Ligne A	3	instruction à l'origine de l'exception
11	0x02C	Emulateur Ligne F	3	instruction à l'origine de l'exception
12	0x030	Point d'arrêt validé	3	instruction suivante du programme
13	0x034	Non affecté (réservé)		sans objet
14	0x038	Erreur de format	1	instruction à l'origine de l'exception
15	0x03C	Interruption non initialisée		instruction suivante du programme
16	0x040	-----		
à	à	Non affectés (réservés)		sans objet
23	0x05C	-----		
24	0x060	Interruption parasite		instruction suivante du programme
25	0x064	Autovecteur Interrupt. Niv. 1		instruction suivante du programme
26	0x068	Autovecteur Interrupt. Niv. 2		instruction suivante du programme
27	0x06C	Autovecteur Interrupt. Niv. 3		instruction suivante du programme
28	0x070	Autovecteur Interrupt. Niv. 4		instruction suivante du programme
29	0x074	Autovecteur Interrupt. Niv. 5		instruction suivante du programme
30	0x078	Autovecteur Interrupt. Niv. 6		instruction suivante du programme
31	0x07C	Autovecteur Interrupt. Niv. 7		instruction suivante du programme
32	0x080	-----		
à	à	Instructions Trap de #0 à #15	4	instruction suivante du programme
47	0x0BC	-----		
48	0x0C0	-----		
à		Non affectés (réservés)		
63	0x0FF	-----		
64	0x100	-----		
à	à	Interruptions vectorisées		instruction suivante du programme
255	0x3FC	-----		

Détails des exceptions :

RESET.... une exception exceptionnelle !

C'est l'exception de niveau prioritaire maximum. Elle est utilisée pour initialiser le système à la mise sous tension ou en cas d'erreur majeure. Elle est causée par la mise à zéro de la broche /RSTI du processeur. Cette mise à zéro est provoquée soit par une mise sous tension, soit par l'action manuelle sur le poussoir « reset ». Tout traitement en cours est avorté, rien n'est sauvegardé.

- 1 Le registre d'état est initialisé: T est mis à 0, S est mis à 1, M est mis à 0
Le masque d'interruption au niveau 7
- 2 Le registre VBR est mis à 0
- 3 L'unité de contrôle des chip-select est initialisé, seul /CS0 est actif.



Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

- 4 Lorsque la broche /RSTI est remise à 1, le processeur « laisse passer du temps » (env. 100000 top d'horloge de CLKIN) , il échantillonne à ce moment là les lignes d0-d7 de son bus des données pour en tirer diverses informations d'initialisation (taille du « port » attaché à CS0, nbr de WS sur ce port, frq. bus...etc)
- Le vecteur 0 est fourni en interne et, exceptionnellement le microprocesseur va chercher deux mots longs dans la zone Programme Superviseur
 - Le premier à l'adresse 0 pour en charger le pointeur de pile .
 - Le second à l'adresse 4 pour initialiser le compteur de programme (PC)
 - Les autres registres (du modèle de programmation) adoptent des valeurs aléatoires. Si une erreur d'accès ou une erreur d'adresse intervient durant l'exécution de la 1ere instruction, une erreur sur erreur intervient. Le processeur se place dans un état « halt ». Il est bloqué. Seul un reset peut le sortir de cet état.

ERREUR D'ACCES

Cette exception est causée par une tentative d'accès en écriture sur un espace mémoire protégé contre ce type d'accès. Ce sont les registres CSMR0-7 (bit8 : WP) qui fixent les « droits » d'accès aux zones mémoires (et/ou périphériques) .

ERREUR ADRESSE

Cette exception survient lorsque le microprocesseur tente de rechercher une instruction à une adresse impaire. Ou bien sur la recherche d'un opérande via l'adressage (ai,xi.w) avec un indice d'échelle non unitaire. Le cycle bus n'est pas exécuté, la procédure est identique à celle de l'erreur d'accès qui est développée dans le paragraphe précédent, cependant le numéro de vecteur est différent 0x0C.

INSTRUCTION ILLEGALE

Une instruction illégale est une instruction dont les bits caractéristiques à l'intérieur du mot de codage de l'instruction (les éventuels mots d'extension ne sont pas concernés) ne correspondent pas à une instruction du jeu d'instructions du processeur coldfire (5307 →).

DIVISION PAR ZERO

Intervient si une instruction de division est réalisée avec l'opérande source (le diviseur) nul. (la division n'est pas réalisée, les registres impliqués ne sont pas altérés)

VIOL DE PRIVILEGE

intervient si une instruction privilégiée est entreprise alors que le bit S du registre d'état vaut 0 (mode user). (exemple l'instruction `move.w #0x2000,sr`)
L'instruction en question n'est pas exécutée.

TRACE

Le mode trace est destiné à aider à la mise au point des programmes en permettant de les exécuter instruction par instruction.
Ce mode utilise le bit T du registre d'état .



Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

Si, au début de l'exécution d'une instruction le bit T est à 1, une exception trace sera générée après l'exécution de l'instruction.

EMULATEUR LIGNES A et F

Le concepteur du microprocesseur coldfire (Motorola) réserve des extensions de son jeu d'instruction en déployant le codage de ces extensions à partir des codes 0xA--- et 0xF---.

C'est déjà le cas de microprocesseur 5307 qui offre une extension du jeu d'instructions initial de la famille avec une unité MAC. Les instructions associées à l'usage de cette unité sont codées 0xA.... Les extensions liées à l'unité FPU des 5407 ->, exploitent les codes 0xF...

Tous les codes non exploités, mais exécutés, conduisent aux deux exceptions émulation ligne A (codes 0xA...) et émulation ligne F (codes 0xF...).

Il est toujours possible, si l'on connaît, les codes que Motorola donne (ou donnera) aux instructions de ses extensions de jeu, d'écrire les programmes équivalents à ces instructions et d'en obtenir l'exécution (« lente ») via l'usage des mécanismes d'exceptions ligne A et/ou F.

POINT D'ARRET

Si les registres de définition des point d'arrêt ont été programmés, et si un point d'arrêt valide est rencontré lors de l'exécution d'un programme, alors c'est cette exception qui est entreprise par le processeur.

ERREUR DE FORMAT

Lors de l'exécution de l'instruction RTE, si le champ « format » empilé sur la pile n'est pas conforme (égal à 4 ou 4 ou 6 ou 7) l'exception erreur de format est entreprise.

INTERRUPTION NON INITIALISEE

Si le vecteur 0x0F (15) est retourné par un périphérique lors d'une procédure d'acquisition du vecteur d'interruption, alors le processeur entreprend cette exception.

Les périphériques Motorola, susceptibles d'être émetteurs d'interruption, mettent leur registre(s) VR à la valeur 0x0F suite à un reset ou une mise sous tension.

INTERRUPTION PARASITE

Si lors de la procédure d'acquiescement d'une interruption, le vecteur délivré conduit à une erreur d'adresse, alors l'exception est entreprise en lieu et place des exceptions erreur d'adresse ou erreur d'accès.

INSTRUCTION TRAP

Associée à l'exécution d'une instruction TRAP #0-15

Le processus de traitement des Trappes est le même que celui des autres exceptions. SR est copié, le Registre d'état actif est positionné S = 1, T = 0. Si le mode trace était validé, l'exception Trap sera traitée d'abord normalement puis le mode Trace sera exécuté. Pour les Trappes, il y en a 16 possibles de Trap 0 à Trap 0xF, le numéro de vecteur est obtenu en ajoutant 32 au N° trap.



Le microprocesseur Cold FIRE 5307

C. GUIRAUDIE

PC et la copie de SR sont sauvés dans la pile superviseur . La valeur de PC est l'adresse de l'instruction suivant celle qui a généré la Trappe. PC est chargé avec l'adresse trouvée puis le traitement du programme Trap commence.

INTERRUPTIONS

Les demandes d'interruption sont effectuées en appliquant un ou plusieurs états actifs sur les broches IRQ 7-5-3-1. 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 ET si le canal associé 7-0 est validé (mis à 0) dans le registre IMR.

Une exception est faite pour l'interruption de niveau 7 qui ne peut être masquée, elle sera toujours acceptée (sauf si invalidée via son bit de masquage de IMR) quelle que soit la valeur introduite dans le masque d'interruption, c'est une interruption non masquable.

Une copie interne de SR 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 :

seul si la source est « qualifiée » d'autovectorisation via la programmation du registre AVR (MBAR + 0x48 ; accès octet)

Via l'acquisition d'un octet vecteur (bus des données d31-24) qui doit fournir le périphérique source d'interruption lors de l'acquiescement que réalise le processeur en mettant à 0 une des broches IRQ 7-5-3-1

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

Si le circuit qui demande l'interruption est de la famille 68000 et que cette demande survient avant son initialisation, le vecteur fourni est 0x0F (voir exception interruption non initialisée)

CONTEXTE SAUVEGARDE LORS D'UNE EXCEPTION

Suite à une exception ou une interruption , les registres PC et SR , le vecteur origine ainsi que des informations complémentaires sont stockés sur la pile suivant le format :

	31	28	27	26	25	18	17	16	15	0	
A7 →	FORMAT		FS [3-2]		VEC			FS [1-0]		Status Register	
+0x04	Program Counter [31 – 0]										

format de pile d'exception/interruption

Format, et champs FS et vecteur sont précisés dans le tableau suivant :



Le microprocesseur *Cold FIRE 5307*

C. GUIRAUDIE

Bits	Field	Description															
31 -28	FORMAT	Format field. Written with a value of (4,5,6,7) by the processor indicating a 2-longword frame format. FORMAT records any longword stack pointer misalignment when the exception occurred															
		<table border="1"> <thead> <tr> <th>A7 at Time of Exception. Bits [1:0]</th> <th>A7 at First Instruction of Handler</th> <th>FORMAT</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>OriginalA7 – 8</td> <td>0100</td> </tr> <tr> <td>01</td> <td>OriginalA7 –9</td> <td>0101</td> </tr> <tr> <td>10</td> <td>OriginalA7 –10</td> <td>0110</td> </tr> <tr> <td>11</td> <td>OriginalA7 –11</td> <td>0111</td> </tr> </tbody> </table>	A7 at Time of Exception. Bits [1:0]	A7 at First Instruction of Handler	FORMAT	00	OriginalA7 – 8	0100	01	OriginalA7 –9	0101	10	OriginalA7 –10	0110	11	OriginalA7 –11	0111
		A7 at Time of Exception. Bits [1:0]	A7 at First Instruction of Handler	FORMAT													
		00	OriginalA7 – 8	0100													
		01	OriginalA7 –9	0101													
10	OriginalA7 –10	0110															
11	OriginalA7 –11	0111															
27-26	FS[3-2]	Fault status. Defined for access and address errors and for interrupted debug service routines															
		0000 Not an access or address errors and for interrupted debug service routine															
		0001 Reserved															
		0010 Interrupt during a debugservice routine for faults other than access error (New in V4)															
		0011 Reserved															
		0100 Error (for example, protection fault) on instruction fetch															
		0101 TLB miss on opword of instruction fetch (New in V4, MMU only)															
		0110 TLB miss on extention word of instruction fetch (New in V4, MMU only)															
		0111 FP access error while executing in emulator mode (New in V4, MMUonly)															
		1000 Error on data write															
		1001 Error on attempted write to write-process space															
		1010 TLB miss on data write (New in V4, MMUonly)															
		1011 Reserved															
1100 Error on data read																	
1101 Attempted read , read-modify-write of protected space (New in V4, MMUonly)																	
1110 TLB miss on data read, or read-modify-write (New in V4, MMUonly)																	
1111 OEP access error while executing in emulator mode (New in V4, MMUonly)																	
25-18	VEC	Vector number. Defines the exception type. It is calculated by the processor for internal faults and is supplied by the peripheral for interrupts.															
17-16	FS[1-0]	See bits 27 – 26															

Le programme d'exception ou d'interruption s'exécute toujours en mode superviseur. pour récupérer le contexte correct sur la pile. (contexte décrit ci-avant) il est nécessaire de mettre en œuvre l'instruction RTE (et non RTS).