

IUT1 GEII2

INFORMATIQUE INDUSTRIELLE

COLDFIRE MCF5272

MODULES PERIPHERIQUES

Rev : 21/05/13

Table des matières

1.Présentation du Coldfire MCF5272.....	5
2.Architecture périphérique du MCF5272.....	5
3.Lecture – écriture des registres des modules périphériques.....	5
3.1.Généralités.....	5
3.2.Accès aux registres.....	6
3.2.1.Accès en assembleur.....	6
3.2.2.Accès en langage C.....	6
4.Module d'entrées – sorties à usage général (GPIO).....	8
4.1.Présentation.....	8
4.2.Principe de fonctionnement.....	8
4.3.Registres.....	8
4.3.1.Registre de configuration (Port X CoNTrol register : PACNT – PDCNT).....	8
4.3.2.Registre de direction (Port X Data Direction Register : PADDR, PBDDR, PCDDR).....	9
4.3.3.Registre de données (Port X Data Register : PADAT, PBDAT, PCDAT).....	9
5.Module d'entrées à drapeaux.....	9
5.1.Principe et caractéristiques.....	9
5.2.Registres associés.....	9
5.2.1.Registre de sélection de front (Programmable Interrupt Transition Register : PITR).....	9
5.2.2.Registre d'état des sources (Interrupt Source Register : ISR).....	10
5.2.3.Drapeaux.....	10
6.Modules timers du MCF5272.....	10
6.1.Présentation.....	10
6.2.Principe de fonctionnement.....	11
6.2.1.Fonction référence.....	11
6.2.2.Fonction capture.....	11
6.3.Schéma de principe des timers.....	11
6.4.Les registres des timers.....	12
6.4.1.Compteurs (Timer Counter : TCN0-TCN3).....	12
6.4.2.Registres de référence (Timer Reference Register : TRR0-TRR3).....	12
6.4.3.Registres de capture (Timer CAPture : TCAP0-TCAP3).....	12
6.4.4.Registres de drapeaux (Timer Event Register : TER0-TER3).....	12
6.4.5.Registres de configuration (Timer Mode Register : TMR0-TMR3).....	13
7.Module de modulation de largeur d'impulsion (MLI ou PWM).....	14
7.1.Présentation.....	14
7.2.Principe de fonctionnement.....	14
7.3.Schéma de principe.....	14
7.4.Registres des modules PWM.....	15
7.4.1.Registre de largeur d'impulsion (Pulse Width register : PWWD0-PWWD2).....	15
7.4.2.Registre de configuration (PWM Control Register PWCR0-PWCR2).....	15
8.Contrôleur d'interruptions.....	16
8.1.Présentation du système d'interruptions.....	16
8.2.Fonctionnement du système d'interruptions vectorisées.....	16
8.2.1.Introduction.....	16
8.2.2.Tables des vecteurs d'exception et d'interruption.....	17
8.3.Reconnaissance d'une interruption.....	17
8.3.1.Masque de priorité d'interruption et registres de priorité (Interrupt Control Register : ICR1-ICR4).....	17
8.3.2.Traitement d'une interruption.....	19
8.4.Registres spécifiques.....	19
8.4.1.Registre de réveil (Programmable Interrupt Wakeup Register : PIWR).....	19
8.5.Tables des vecteurs.....	20
8.5.1.Table des vecteurs d'interruption du MCF5272.....	20
8.5.2.Table des vecteurs d'exception.....	21
9.Tableaux récapitulatifs des registres des modules périphériques et du contrôleur d'interruptions.....	22
9.1.Registres du contrôleur d'interruptions.....	22
9.2.Registres des modules PWM.....	23
9.3.Registres des timers.....	24
9.4.Registres de configuration des broches multifonctions.....	25
9.5.Registres des modules GPIO.....	25

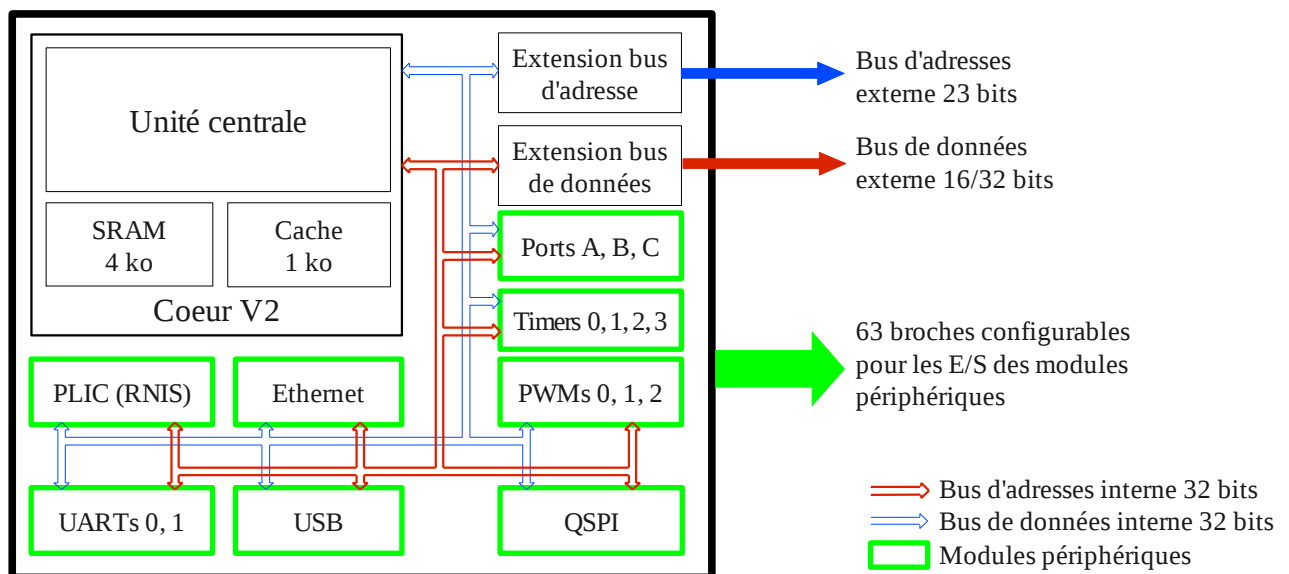
1. Présentation du Coldfire MCF5272

Le Motorola ColdFire 5272 est un microcontrôleur RISC¹ 32 bits. En tant que microcontrôleur, il intègre donc une unité centrale (qu'on appellera aussi cœur ou core), de la RAM, et de nombreux périphériques d'entrées/sorties permettant d'envisager un très grand nombre de modes de communication avec son environnement. En revanche, contrairement au 5282, il ne possède pas de ROM (EEPROM Flash) et ne peut donc pas fonctionner en "stand alone". Il sera donc nécessairement accompagné par une EEPROM et éventuellement par de la RAM.

Le cœur ("core" - noyau - en anglais) est hérité de l'architecture de la famille 68000 de Motorola, conçue à la fin des années 70. Son unité arithmétique et logique et ses registres de manipulation des données et des adresses sont d'une taille de 32 bits.

La famille Coldfire se décompose en trois catégories de cœurs (V2, V3, V4(e)). Le MCF5272 a un cœur V2. Pour chaque type de cœur existe une grande variété de composants comprenant un nombre important de modules périphériques. Le choix du composant pour une application industrielle donnée est donc en grande partie guidé par les modules périphériques intégrés au circuit.

2. Architecture périphérique du MCF5272



3. Lecture – écriture des registres des modules périphériques

3.1. Généralités

Dans un Colfire, (à l'inverse des systèmes à base d'Intel x86), les registres des modules périphériques sont considérés comme des cases mémoires. Ils sont donc dans le même plan mémoire que la RAM et la ROM et occupent une place qui ne doit pas se chevaucher avec la RAM ou la ROM. Le Coldfire possède un décodeur d'adresses interne au composant, qui génère des signaux de "chip select" pour sélectionner la RAM, la ROM, les

1 RISC - Reduced Instruction Set Computer : processeur à jeu d'instruction réduit. Ce type de processeur exécute normalement une instruction par cycle d'horloge. Sur le MCF5272 cadencé à 66 Mhz, chaque instruction est exécutée en 60 ns sauf quelques unes qui ne respectent pas cette règle (exemple : division)

modules périphériques. Ce décodeur d'adresses interne est de plus configurable afin de permettre différentes organisations du plan mémoire.

3.2. Accès aux registres

Les registres des modules périphériques ont 3 tailles possibles : 8 bits, 16 bits, 32 bits. L'écriture de ces registres doit donc se faire de manière très prudente car une écriture au format long (32 bits) sur un registre de taille inférieure entraînera soit une erreur bus, soit la modification d'autres registres.

On présente ici l'accès aux registres en assembleur et en C, sous différentes écritures.

3.2.1. Accès en assembleur

L'accès en langage assembleur se fait avec l'instruction `move` sous ses 3 déclinaisons `move.b`, `move.w`, `move.l` pour respectivement des accès 8, 16, 32 bits. Il est nécessaire de passer par l'intermédiaire d'un registre de données puisque l'adressage immédiat n'est pas supporté lors de l'accès à une case mémoire.

Ci-dessous un exemple de configuration du port A en écriture (grâce au registre `PADDR`), puis de l'écriture du nombre 16 bits "ABCD" sur le port A (`PADAT`) :

```
move.w    #0xFFFF,%d0
move.w    %d0,0x10000084 //0x10000084 est l'adresse de PADDR
move.w    #0xABCD,%d0
move.w    %d0,0x10000086 //0x10000086 est l'adresse de PADAT
```

3.2.2. Accès en langage C

En langage C, la seule façon d'accéder à une case mémoire d'une adresse donnée, est d'utiliser un pointeur. Afin d'éviter la génération d'un code erroné à cause d'optimisations abusives du compilateur C, on utilisera toujours un type `volatile`. La ligne suivante est donc équivalente aux 2 premiers `move` de l'exemple précédent :

```
*(volatile unsigned short *) (0x10000084) = 0xffff;
```

En fonction des 3 tailles de registre, voici les 3 types à utiliser :

- ◆ 8 bits (octet) : `volatile unsigned char`
- ◆ 16 bits (mot) : `volatile unsigned short`
- ◆ 32 bits (long mot) : `volatile unsigned int`

Afin de simplifier les écritures, le fichier `mcf5272.h` définit les offsets de tous les registres des modules périphériques du MCF5272. Par exemple, on trouve dans le fichier `mcf5272.h` :

```
#define PADDR 0x10000084
#define PADAT 0x10000086
```

Les noms des registres sont ceux utilisés dans la suite du document, en conformité avec la datasheet du MCF5272. La ligne de code précédente devient alors :

```
#include "mcf5272.h" //à insérer au début du fichier
*(volatile unsigned short *)PADDR = 0xffff;
```

Pour compacter l'écriture du code C, le fichier `mcf5272.h` définit les macros suivantes : `io_8(adresse)`, `io_16(adresse)`, `io_32(adresse)`. Par exemple, on trouve dans le fichier `mcf5272.h` :

```
#define io_16(ADDR) *((volatile unsigned short *) (ADDR))
```

La ligne de code précédente devient alors :

```
io_16(PADDR) = 0xffff;
```

4. Module d'entrées – sorties à usage général (GPIO)

4.1. Présentation

Par "entrées – sorties à usage général" (GPIO : "General Purpose Inputs - Outputs"), on entend toutes les broches du composant qui permettent l'application ou la lecture d'un niveau logique, indépendamment des modules spécifiques.

Le Coldfire MCF5272 possède en tout 3 ports 16 bits (A, B, C) d'entrées – sorties utilisables en GPIO. Tous utilisent des broches multifonctions, et requièrent l'écriture d'un registre de configuration avant d'être utilisés.

- ♦ Le port A est bien adapté à un usage général car il partage ses broches avec des modules peu utilisés (USB, QSPI, PLIC). Le bit 15 pose cependant problème car il est utilisé par défaut pour le signal $\overline{INT6}$ qui sert le bouton "ABORT" de la carte Cobra. La modification de la configuration du système d'interruption est donc nécessaire pour utiliser PA15.
- ♦ Le port B, à l'exception de quelques bits, utilise des broches multifonctions pour la plupart utilisées par les modules de communication (timer0, UART 0 et 1, Ethernet...) ; l'UART ou le module ethernet étant fondamentaux pour le fonctionnement du debug, seuls quelques bits sont utilisables pour un usage général.
- ♦ Le port C utilise des broches multifonctions partagées avec les 16 bits de poids faible de l'extension du bus de données ; celui-ci étant utilisé par la SDRAM de la carte, le port C est de fait inutilisable pour un usage général.
- ♦ En plus de ces 3 ports, le Coldfire MCF5272 possède 8 broches à usage multiple mais réservé à certains modules. Ces broches sont regroupées sous le nom de port D et ne peuvent pas être utilisées pour un usage général. On trouve notamment les signaux PWMOUT1 et 2, TIN1, TOUT1...

Par défaut, au reset, les ports sont configurés en GPIO. Mais sur le kit utilisé, le debug configure les ports de façon à utiliser les liaisons séries et ethernet notamment. Avant d'utiliser un port en GPIO, il est donc nécessaire de modifier le registre de configuration associé à ce port.

4.2. Principe de fonctionnement

L'écriture ou la lecture d'un port d'entrées – sorties se fait par l'écriture ou la lecture des données dans un registre de données. Le sens de chaque broche d'E/S est défini par un registre de direction ; par défaut, les broches sont configurées en entrées. Lorsque l'on souhaite écrire sur une sortie, il est préférable de commencer par écrire le registre de données puis celui de direction afin d'éviter l'apparition d'un état transitoire indésirable sur la sortie.

Sur le schéma ci-contre, on voit que le coeur d'un bit de port GPIO peut être considéré comme piloté par 2 bascules D : une qui définit le sens du port (PADDR0) et une qui mémorise la valeur (PADAT0) à présenter en sortie (lorsque le port est configuré en sortie par PADDR0).

Le signal D0 est le bit 0 du bus de données du processeur. Le signal PA0 est la broche 0 du port A.

Les signaux DAT, DDR, Lddr et Ldat sont générés à partir du bus d'adresse et de contrôle du processeur (à partir du décodage de l'adresse, et des signaux R/W et DataStrobe).

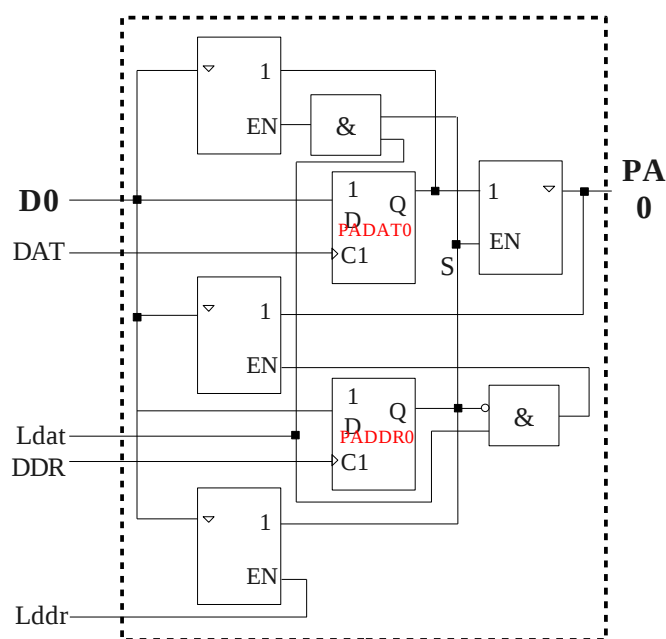


Schéma de principe d'un bit d'un port GPIO

4.3. Registres

Comme dans la majorité des micro-contrôleurs, la plupart des broches du MCF5272 peuvent être configurées pour plusieurs fonctions différentes. Ceci est réalisé grâce à un multiplexeur interne configurable.

4.3.1. Registre de configuration (Port X CoNTrol register : PACNT – PDCNT)

Le registre de configuration définit l'usage fait de chaque broche multifonction. Sa taille est 32 bits pour les ports A et B. Sa taille est 16 bits pour le port D (8 broches) qui n'est pas utilisable en GPIO. Pour chaque broche d'entrée – sortie, 2 bits définissent sa fonction, ce qui autorise jusqu'à 4 fonctions.

Il est important de noter que sur le kit Cobra, la plus grande partie du port B est utilisé pour les signaux de l'UART0 (COM1) et du module Ethernet et est donc indisponible pour des GPIO.

Le port C ne possède pas de registre de configuration ; sur le kit Cobra, il n'est pas utilisable en GPIO car il sert le mot de poids faible de l'extension du bus de données. Pour les ports A et B, l'organisation est la suivante :

Bits PACNT	31-30	29-28	27-26	25-24	23-22	21-20	19-18	17-16	15-14	13-12	11-10	9-8	7-6	5-4	3-2	1-0
Broche PA	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

Pour utiliser une broche en GPIO, il faut que les bits PxCNT correspondant soient à "00". Le détail des autres fonctions des broches et leurs codes sont donnés dans les tableaux récapitulatifs à la fin de l'ouvrage.

4.3.2. Registre de direction (Port X Data Direction Register : PADDR, PBDDR, PCDDR)

Il y a un registre de direction pour chaque port utilisable en GPIO. Chaque bit du registre de direction définit le sens de la broche correspondante (si configurée en GPIO). Sa taille est donc de 16 bits.

'0' : broche correspondante en entrée.

'1' : broche correspondante en sortie.

Au reset, les broches GPIO sont configurées en entrée. Lorsqu'une broche passe d'entrée en sortie, le niveau logique appliqué est le dernier contenu dans le registre de données avant le passage en entrée.

Exemple en C de configuration de tout le PORTA en sortie :

```
io_16(PADDR) = 0xFFFF; //16 bits à '1'
```

4.3.3. Registre de données (Port X Data Register : PADAT, PBDAT, PCDAT)

Le registre de données (16 bits) sert indifféremment à lire ou écrire une broche GPIO. Il n'a pas d'effet si la broche correspondante n'est pas configurée en GPIO.

Lors d'une lecture, le registre contient les valeurs instantanées des niveaux logiques présents sur les broches configurées en entrées. Les broches configurées en sorties présentent en permanence le niveau logique contenu dans le registre de données.

Exemple en C de modification du bit 7 du PORTA à '1', sans modification des autres bits :

```
io_16(PADAT) = io_16(PADAT)|0x0080; //masquage pour mise à '1' du bit n°7
```

Exemple en C de modification du bit 9 du PORTA à '0', sans modification des autres bits :

```
io_16(PADAT) = io_16(PADAT)&0xFDFF; //masquage pour mise à '0' du bit n°9
```

5. Module d'entrées à drapeaux

5.1. Principe et caractéristiques

Le MCF5272 possède six entrées logiques à drapeau (INT1 à INT6). Ces lignes ne peuvent être utilisées qu'en entrées et ne possèdent donc pas de registre de sens comme les autres ports. En revanche, une structure logique additionnelle permet la détection de fronts montants ou descendants sur chaque entrée. La détection d'un tel évènement provoque la mise à 1 d'un bit de drapeau (ou "flag").

5.2. Registres associés

5.2.1. Registre de sélection de front (Programmable Interrupt Transition Register : PITR)

Ce registre définit le type de front à détecter sur les entrées externes $\overline{INT1}$ à $\overline{INT6}$.

D'une taille de 32 bits, seuls les bits 5, 6, 28, 29, 30, 31 sont utilisés :

N° bit	31	30	29	28	27	7	6	5	4	0
Champ	INT1	INT2	INT3	INT4	-	-	-	INT5	INT6	-

0 : détection d'un **front descendant**, 1 : détection d'un **front montant**.

5.2.2. Registre d'état des sources (Interrupt Source Register : ISR)

Le registre **ISR** permet de lire la valeur instantanée des entrées \overline{INTx} . Il sert à lire le niveau logique des lignes externes $\overline{INT1}$ à $\overline{INT6}$, indépendamment des drapeaux et de leur configuration. Ceci permet d'utiliser ces entrées comme un port GPIO classique, mais en entrée seulement.

La position des bits des entrées externes dans **ISR** est la même que dans **PITR**.

5.2.3. Drapeaux

Les bits de drapeaux sont contenus dans les registres **ICR1** et **ICR4**. Ces registres contiennent aussi la configuration du contrôleur d'interruption, lorsque ces entrées sont utilisées pour générer une interruption (voir §8.3.1)

Registre ICR1 : (partie 16 bits de poids fort uniquement)

N° bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICR1	INT1PI	INT1IPL	INT2PI	INT2IPL	INT3PI	INT3IPL	INT4PI	INT4IPL	-	-	-	-	-	-	-	-

Les bits de drapeaux se nomment \overline{INTxPI} . Les flags $\overline{INT5PI}$ et $\overline{INT6PI}$ se trouvent dans le registre **ICR4** (voir tableau récapitulatif au §9.1)

Lorsque l'on a testé qu'un drapeau est "levé" dans un programme, il est nécessaire de "l'abaisser", c'est à dire le remettre à '0', pour être prêt à détecter un nouvel évènement.

Remarque importante : Comme pour tous les autres flags du MCF5272 (voir timers et autres modules), LA REMISE À ZÉRO DES FLAGS \overline{INTxPI} SE FAIT **EN ECRIVANT UN '1' DESSUS**. L'écriture d'un '0' sur les bits PI n'a tout simplement aucun effet. Ceci permet d'éviter l'utilisation fastidieuse des masques de mise à zéro.

Exemple en C de remise à '0' du flag $\overline{INT2PI}$ de l'entrée $\overline{INT2}$:

```
io_32(ICR1) = 0x08000000; //masquage inutile
```

Attention : les bits $\overline{INTxIPL}$ servent à configurer le déclenchement d'interruptions par ces entrées. Il est important de laisser à 0 ces bits là lors du RAZ du bit PI correspondant, sous peine de déclencher des interruptions non souhaitées (ceci sera étudié au §8.3.1).

6. Modules timers du MCF5272

6.1. Présentation

Les timers permettent au microprocesseur de gérer des évènements liés à une durée. Le cœur d'un timer est un compteur. Un type particulier de timer est le chien de garde qui permet de surveiller l'absence de plantage du microprocesseur.

Le ColdFire 5272 comporte 4 timers identiques et un timer chien de garde. Les caractéristiques principales des 4 timers sont :

- ♦ compteur 16 bits,

- ♦ période maximale de 4s (résolution 15ns avec horloge système à 66MHz),
- ♦ prédiviseur configurable de 1 à 256,
- ♦ entrée d'horloge externe,
- ♦ fonction de capture,
- ♦ sortie de résultat de comparaison,
- ♦ 5 registres dédiés à chaque timer : TMRn, TRRn, TCAPn, TCNn, TCRn ($0 \leq n \leq 3$).

6.2. Principe de fonctionnement

Le cœur de chaque timer est constitué d'un compteur 16 bits (TCNn) pouvant donc compter de 0 à 65535. L'horloge du compteur est issue d'un prédiviseur de rapport compris entre 1 et 256. L'horloge alimentant le prédiviseur peut être externe (entrées TINn), ou issue de l'horloge principale du processeur (après division ou non par 16).

6.2.1. Fonction référence

Le timer peut être configuré pour compter jusqu'à atteindre une référence contenue dans le registre de référence TRRn. Une fois la référence atteinte, et en fonction du bit FRR ("Free Run/Restart", registre TMRn), le compteur peut continuer à compter librement, ou bien être remis à 0 automatiquement.

Lorsque la référence est atteinte, le drapeau REF (registre TERn) passe à 1. Une interruption est demandée si le bit ORI ("Output Reference Interrupt", registre TMRn) est à 1.

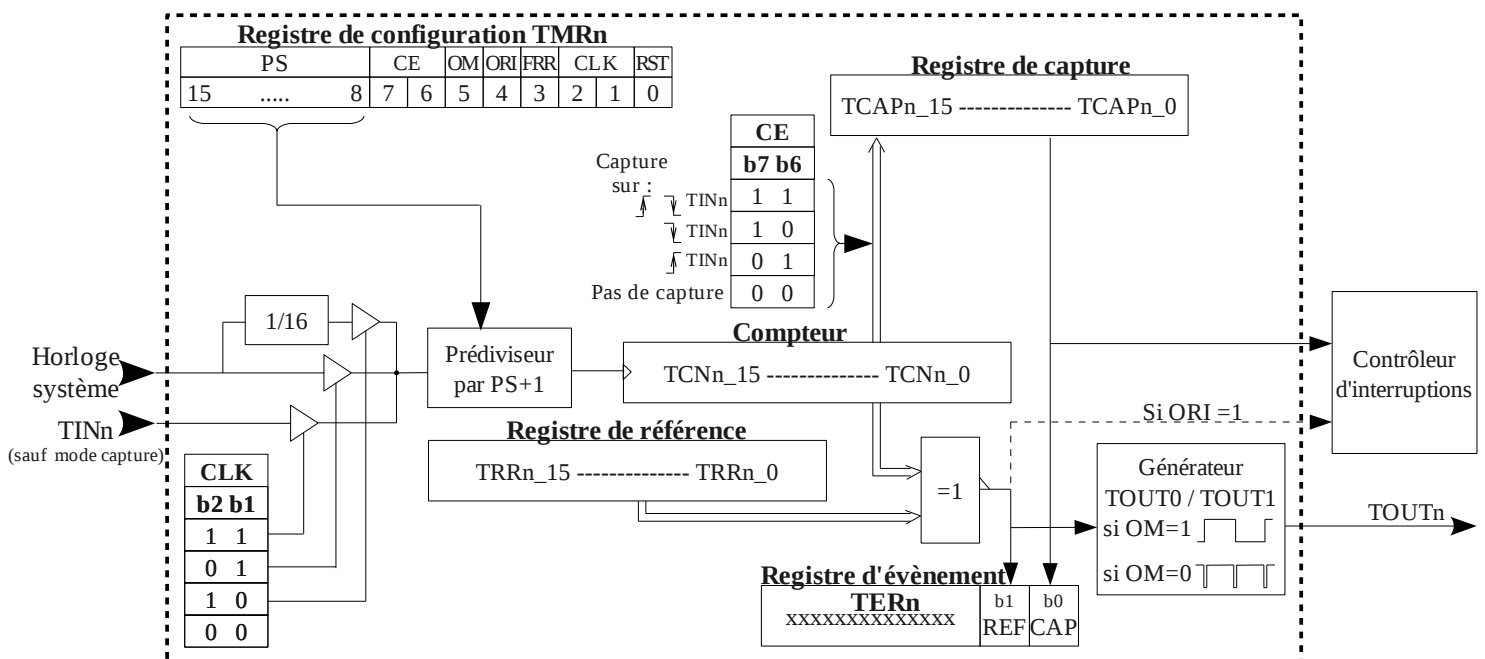
Les timers 0 et 1 possèdent une sortie TOUTn qui peut être configurée (bit OM, "Output Mode", registre TMRn) pour basculer à chaque fois que la référence est atteinte, ou pour délivrer une impulsion (15ns à 66MHz), active au niveau bas lorsque la référence est atteinte.

6.2.2. Fonction capture

Les timers possèdent un registre TCAPn utilisé pour capturer la valeur du compteur. La capture est déclenchée par un événement extérieur sur la broche TINn (timers 0 et 1) ou UARTn_RxD (timers 2 et 3). Les bits CE ("Capture Edge", registre TMRn) activent ce mode et déterminent le front sur le signal d'entrée déclenchant la capture.

La capture déclenche la mise à 1 du drapeau CAP (registre TERn) et une demande d'interruption.

6.3. Schéma de principe des timers



- ◆ Les timers 2 et 3 ne comportent pas de sortie Tout ni d'entrée externe d'horloge TINn ; seule l'horloge système est utilisée.
- ◆ En mode capture, les timers 2 et 3 utilisent les entrées URT0_RxD et URT1_RxD à la place de TIN0/1 pour les timers 0 et 1.
- ◆ La sortie TOUT0 est câblée sur une broche multifonction. Les bits PBCNT15-14 doivent être fixés à "01" pour que cette broche soit affectée au timer. Les entrées TIN1, URT1_RxD et la sortie TOUT1 sont câblées sur des broches multifonctions. Les bits respectifs PDCNT15-14, PDCNT3-2 et PDCNT13-12 doivent être fixés à "10" pour que ces broches soient affectées aux timers. L'utilisation de TIN1, URT1_RxD, TOUT1 est respectivement incompatible avec celle de PWM2, DIN0, et PWM1.

6.4. Les registres des timers

6.4.1. Compteurs (Timer Counter : TCN0-TCN3)

Le compteur est constitué d'un registre 16 bits. La lecture de ce registre permet de connaître la valeur instantanée du compteur du timer. L'écriture dans ce registre provoque sa remise à 0, quelle que soit la valeur écrite.

6.4.2. Registres de référence (Timer Reference Register : TRR0-TRR3)

Le registre de référence est d'une taille de 16 bits. Le contenu du registre TRRn est comparé au compteur TCNn correspondant.

Lorsque l'égalité survient :

- ◆ la sortie Tout du timer (seulement pour timers 0 et 1) bascule ou génère une impulsion au niveau bas (voir bit OM de TMRn),
- ◆ une interruption est générée si elle est autorisée (bit ORI de TMRn),
- ◆ le drapeau REF est mis à 1 (registre TERn).

6.4.3. Registres de capture (Timer CAPture : TCAP0-TCAP3)

Le registre de capture (16 bits) permet de capturer la valeur instantanée du compteur lors d'un événement sur l'entrée TIN0-TIN1 pour les timers 0 et 1 et sur l'entrée RxD des UART 0 et 1 pour les timers 2 et 3.

Le mode capture est activé par les bits CE du registre TMRn. Lorsqu'un front (montant et/ou descendant, en fonction des bits CE) survient sur cette entrée :

- ◆ la valeur du registre TCNn est copiée dans TCAPn,
- ◆ une interruption est générée,
- ◆ le drapeau CAP est mis à 1 (registre TERn).

6.4.4. Registres de drapeaux (Timer Event Register : TER0-TER3)

Le registre de drapeaux est d'une taille de 16 bits mais seuls les 2 bits de poids faible sont implémentés. Le bit 1 de TERn est le drapeau REF (Output REFERENCE event), le bit 0 est le drapeau CAP (CAPture event). Les bits REF et CAP sont respectivement mis à 1 par l'apparition d'une égalité entre TCNn et TRRn et par l'apparition d'une capture.

Les drapeaux **doivent être remis à 0 par le programmeur** pour que le timer cesse de demander une interruption au contrôleur d'interruptions (acquiescement de la demande). **La mise à 0 se fait par l'écriture d'un 1** sur le bit correspondant. L'écriture d'un '0' n'a pas d'effet. Cette technique permet d'éviter l'utilisation fastidieuse de masques. Les 2 drapeaux peuvent être effacés simultanément.

Les drapeaux sont toujours mis à 1 par un événement, même si l'interruption correspondante est désactivée (bits ORI et CE de TMRx).

Exemple en C de remise à '0' du flag REF du timer2 :

```
io_16(TER2) = 2;          //masquage inutile
```

Exemple en C de remise à '0' des 2 flags du timer0 :

```
io_16(TER0) = 3;          //masquage inutile
```

6.4.5. Registres de configuration (Timer Mode Register : TMR0-TMR3)

Le registre TMR (16 bits) permet de configurer le fonctionnement du timer.

Bits	Nom	Description
15-8	PS	"PreScaler". Cet octet définit le rapport de division du pré-diviseur d'horloge. Le rapport de division est égal à PS+1. Si PS = 0 : l'horloge système (66MHz) est divisée par 1. PS = 255 : division par 256.
7-6	CE	"Capture Edge". Valide le mode capture et définit le front actif du signal de capture. 00 Capture et interruption désactivées 01 Capture sur front montant et interruption générée lors d'un événement de capture 10 Capture sur front descendant et interruption générée lors d'un événement de capture 11 Capture sur n'importe quel front et interruption générée lors d'un événement de capture
5	OM	"Output Mode". (Utilisé sur TMR0 et TMR1 seulement, réservé sur TMR2 et TMR3) Fonctionnement de la broche TOUTn lorsque TCNn=TRRn : 0 Génération d'une Impulsion active au niveau bas (durée d'un cycle d'horloge système soit 15ns) 1 Inversion de TOUTn Au reset, TOUTn est à 1 mais n'est pas disponible en sortie tant que PBCNT7 (pour TOUT0) ou PDCNT6 (pour TOUT1) n'est pas configuré.
4	ORI	"Output Reference Interrupt enable". 0 Pas d'interruption lorsque TCNn=TRRn. Le passage à 1 du drapeau REF ne provoque pas d'interruption. 1 Interruption lorsque TCNn=TRRn activée. Le passage à 1 du drapeau REF provoque une interruption.
3	FRR	"Free Run/Restart" 0 Comptage libre : le compteur du timer continue de s'incrémenter après avoir atteint TRRn 1 Reset compteur : le compteur repart de 0 dès qu'il a atteint TRRn
2-1	CLK	"input ClocK source" : source d'horloge du timer 00 Pas d'horloge, comptage arrêté. 01 Horloge système (66MHz) 10 Horloge système divisée par 16 11 TINn est la source d'horloge (sur front descendant) du timer correspondant. La fréquence maximale de TINn est égale à la moitié de celle de l'horloge système (33MHz). Cette source d'horloge n'est pas disponible pour les timers 2 et 3.
0	RST	"ReSeT timer" 0 Une transition de 1 à 0 remet à 0 le timer (registre TCNn). Tant que RST reste à 0, le compteur reste bloqué à 0. 1 Timer activé.

7. Module de modulation de largeur d'impulsion (MLI ou PWM)

7.1. Présentation

Le module PWM (Pulse Width Modulation) permet de générer des signaux carrés de fréquence et de rapport cyclique variables. Il permet par exemple de réaliser à faible coût un convertisseur numérique/analogique, à condition d'utiliser un filtre passe-bas adéquat à sa sortie.

Ce module a pour caractéristiques :

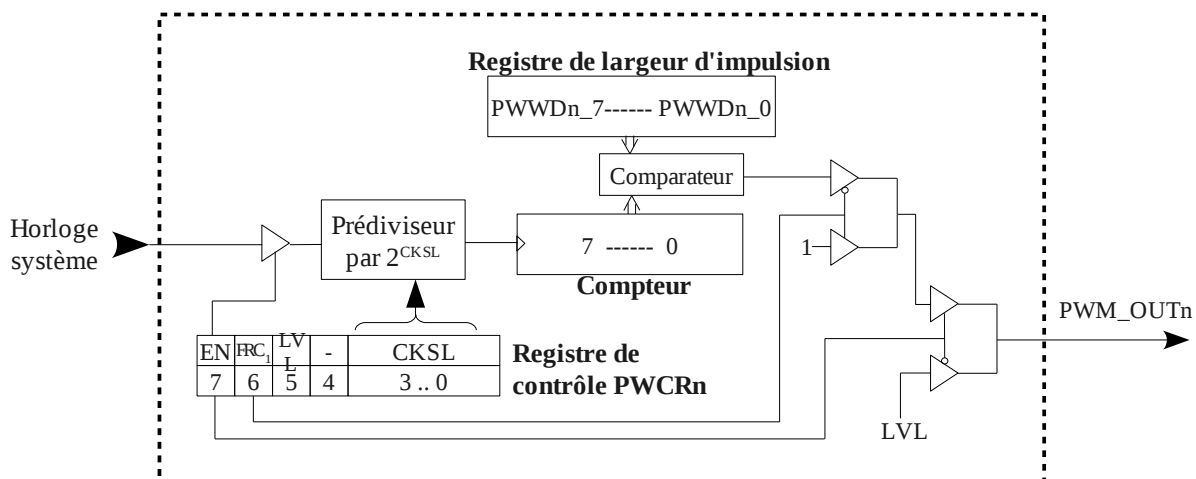
- ♦ trois modules identiques comportant chacun 2 registres de configuration
- ♦ résolution du rapport cyclique : 1/256 (registre de largeur d'impulsion de taille 8 bits)
- ♦ fréquence du signal comprise entre 7,9Hz et 258kHz (avec horloge système à 66MHz)
- ♦ prédiviseur d'horloge de rapport configurable entre 1 et 32768
- ♦ broches multifonctions pour les sorties PWM_out1 et PWM_out2.

7.2. Principe de fonctionnement

Ce module est constitué d'un compteur 8 bits associé à un registre de largeur d'impulsion (PWWDn). Un comparateur de ces registres commande la sortie PWM : lorsque la valeur du compteur est inférieure ou égale au contenu de PWWDn, la sortie est à 1. Lorsque le compteur dépasse PWWDn, la sortie bascule à 0. Le compteur se réinitialise automatiquement à 0 lorsqu'il a atteint 255.

Un prédiviseur d'horloge configuré dans le registre PWCRn permet de régler la fréquence du signal de sortie.

7.3. Schéma de principe



Notes : Les sorties PWM_out1 et 2 sont câblées sur des broches multifonctions. Afin d'utiliser les modules PWM 1 et 2, il est impératif de configurer à "01" les bits 13-12 du registre PDCNT pour le module 1, et les bits 15-14 pour le module 2. L'utilisation de PWM_out1 est incompatible avec celle de TOUT1. L'utilisation de PWM_out2 est incompatible avec celle de TIN1.

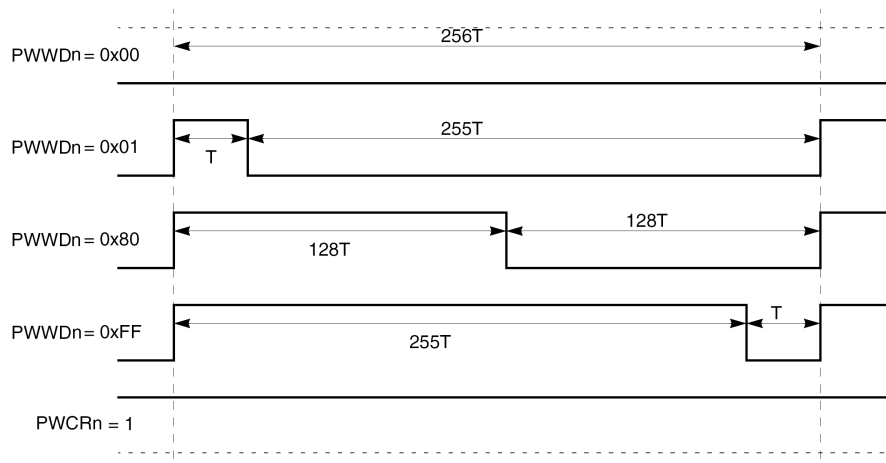
7.4. Registres des modules PWM

7.4.1. Registre de largeur d'impulsion (Pulse Width register : PWWD0-PWWD2)

Ce registre a pour taille 8 bits. Le rapport cyclique est défini comme $PWWDn/256$. Sa valeur est donc comprise entre 0/256 et 255/256.

Ce registre est constitué d'un double buffer qui permet sa modification en cours de fonctionnement. La nouvelle valeur ne sera prise en compte que lorsque le compteur rebouclera de 255 à 0.

Exemples de signaux sur PWMoutn pour différentes valeurs de PWWDn :



7.4.2. Registre de configuration (PWM Control Register PWCR0-PWCR2)

Ce registre a pour taille 8 bits. Il configure le fonctionnement du module PWM correspondant.

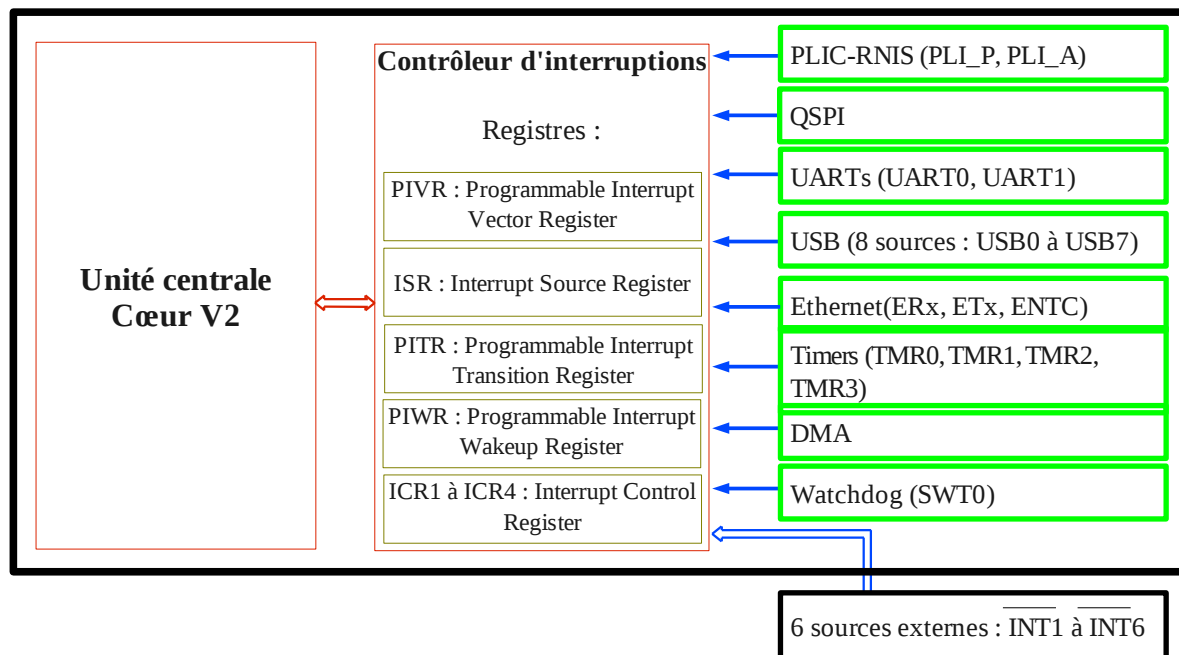
De même que pour le registre PWWDn, des modifications sur ce registre ne seront prises en compte qu'à la fin du cycle de comptage en cours, à l'exception du bit EN (enable) qui permet une désactivation immédiate de la sortie, quel que soit l'état actuel du module. Par exemple, si on souhaite modifier immédiatement (sans attendre la fin du cycle de comptage en cours) la valeur du prédiviseur, il faut désactiver la sortie (mise à 0 de EN), modifier le prédiviseur, puis réactiver la sortie.

Bits	Nom	Description
7	EN	"Enable". Active ou désactive le module PWM. 0 Désactive le module. Le module est en mode économie d'énergie et le prédiviseur ne compte pas. La sortie est bloquée dans l'état défini par le bit LVL. 1 Module PWM activé.
6	FCR1	Force la sortie au niveau haut. 0 Fonctionnement normal de la PWM (valeur par défaut au reset) 1 Sortie forcée au niveau haut. FCR1 n'est pris en compte qu'au rebouclage du compteur de PWM. EN est prioritaire sur FCR1, ce qui signifie que si LVL=0 et que EN=0 (PWM désactivée) alors la sortie sera au niveau bas, même si FCR1=1.
5	LVL	"disable LeVeL". Définit le niveau logique de la sortie lorsque le module est désactivé. 0 Sortie au niveau bas lorsque EN=0. 1 Sortie au niveau haut lorsque EN=0.
4	-	Réservé, devrait être à 0.
3-0	CKSL	Définit le rapport de division du pré-diviseur d'horloge pour régler la période du signal PWM. Le rapport de division est égal à 2^{CKSL} : CKSL[3..0] Rapport de division 0000 1 0001 2 0010 4 1111 32768

8. Contrôleur d'interruptions

8.1. Présentation du système d'interruptions

Le Coldfire MCF5272 possède en tout 28 sources d'interruptions matérielles dont 6 signaux d'interruptions externes nommés $\overline{\text{INT1}}$ à $\overline{\text{INT6}}$. Les 22 autres sources d'interruptions sont issues des modules périphériques intégrés au MCF5272. La figure ci-dessous résume les 28 sources possibles :



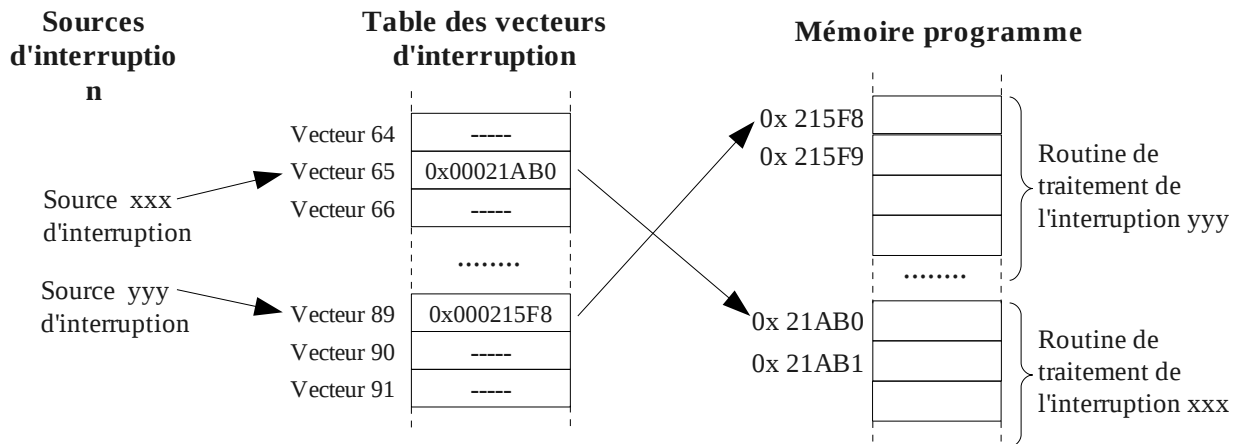
8.2. Fonctionnement du système d'interruptions vectorisées

8.2.1. Introduction

Un système d'interruptions vectorisées permet d'associer à chaque source d'interruption une routine d'interruption. On appelle routine d'interruption un petit programme effectuant le traitement conséquent à l'interruption.

Sur un système vectorisé, chaque source d'interruption est associée à un *vecteur d'interruption*. Un vecteur d'interruption est un pointeur pointant sur le début de la routine d'interruption. Les vecteurs d'interruptions sont stockés dans une table nommée *table des vecteurs d'interruption*.

Le schéma ci-dessous illustre ce fonctionnement, pour deux sources xxx et yyy d'interruption.



Sur cet exemple, on voit que la table d'interruption fait le lien entre la source d'interruption xxx et le morceau de programme (situé à l'adresse 0x21AB0) à exécuter lorsque l'interruption survient.

8.2.2. Tables des vecteurs d'exception et d'interruption

A chacune des 28 sources d'interruption, est attribué un n° de vecteur d'interruption dans une table de 256 vecteurs, utilisée aussi pour les vecteurs d'exception (une interruption est considérée comme une exception matérielle).

Les vecteurs n° 0 à 63 sont utilisés pour les exceptions logicielles (voir table au §8.5.2). La table des vecteurs d'interruption (32 vecteurs) se trouve donc par défaut placée à partir du vecteur n° 64.

Ainsi, l'interruption INT1 est associée au vecteur n° 65, INT2 au vecteur n° 66, et ainsi de suite jusqu'au vecteur n° 95.

On trouvera au §8.5.1 la table des vecteurs d'interruption et au §8.5.2 la table des vecteurs des exceptions logicielles.

8.3. Reconnaissance d'une interruption

A chaque source d'interruption est associé un niveau de priorité. Ceci permet de traiter les interruptions dans un ordre de priorité, au cas où plusieurs interruptions surviennent simultanément. Une interruption peut interrompre le programme de traitement d'une autre, si elle est de priorité supérieure.

8.3.1. Masque de priorité d'interruption et registres de priorité (Interrupt Control Register : ICR1-ICR4)

Le registre SR du cœur du Coldfire contient 3 bits $I_2I_1I_0$ qui définissent un masque de priorité d'interruption : le programme en cours d'exécution ne peut être interrompu que si une interruption est de priorité strictement supérieure au masque $I_2I_1I_0$.

La priorité de chaque source d'interruption est définie dans les registres ICR1 à ICR4 du contrôleur d'interruptions. La description complète de ces registres est donnée dans les tableaux récapitulatifs du §9.

Registre ICR1 :

N° bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICR1	INT1PI	INT1IPL			INT2PI	INT2IPL			INT3PI	INT3IPL			INT4PI	INT4IPL		

N° bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR1	TMR0PI	TMR0IPL			TMR1PI	TMR1IPL			TMR2PI	TMR2IPL			TMR3PI	TMR3IPL		

Pour chaque source d'interruption, les bits xxxxIPL (Interrupt Priority Level) définissent son niveau de priorité. Il existe donc 7 niveaux de priorité de 1 à 7, plus le niveau 0 qui correspond à une désactivation de la source d'interruption correspondante. Si plusieurs interruptions de niveau IPL identique sont actives simultanément, le contrôleur d'interruption place en file d'attente les demandes, dans l'ordre des champs des registres ICRn, avec INT1 la plus prioritaire et SWTO la moins prioritaire.

Exemple de prise en compte des interruptions pour un masque égal à 3 :

Registre d'état SR du cœur du processeur :

T	-	S	M	-	I2	I1	I0	-	-	-	X	N	Z	V	C
---	---	---	---	---	----	----	----	---	---	---	---	---	---	---	---

Niveau de l'interruption		Bits IPL de l'interruption			Etat du masque		
		IPL2	IPL1	IPL0	I2	I1	I0
Niveau le + élevé	7	1	1	1			
Interruptions autorisées	6	1	1	0			
	5	1	0	1			
	4	1	0	0			
Interruptions masquées	3	0	1	1	0	1	1
	2	0	1	0			
Niveau le plus faible	1	0	0	1			
Interruption désactivée	0	0	0	0			

Note : lorsqu'une interruption **externe** INTx est de niveau 7, elle est non-masquable. Cela signifie que même si $I_2I_1I_0=7$, l'interruption sera reconnue. Ceci est un cas particulier non décrit par le tableau ci-contre et qui ne concerne que les sources externes.

Exemple :
masque = 3

Pour chaque source d'interruption, le bit xxxxPI de ICRn (Pending Interrupt), en lecture, est un drapeau (ou bit "flag") actif au niveau 1, signalant la présence d'une demande d'interruption ; en lecture, il ne sert vraiment que pour les entrées INTx, car pour les autres modules, ces flags ne sont qu'une recopie hardware des flags du module correspondant (exemple : bits CAP et REF du timer 0 recopiés dans le bit TMR0PI. En écriture, ce bit sert de validation de l'écriture des bits IPL.

Ecriture des bits IPL :

Pour écrire les 3 bits IPL d'une interruption donnée, il faut simultanément écrire un 1 sur le bit PI correspondant pour valider l'écriture. Ceci permet d'éviter l'utilisation fastidieuse d'un masque : il suffit d'écrire un 0 sur tous les autres bits PI du registre et ainsi les autres IPL ne seront pas modifiés.

Exemple en C de configuration du niveau de priorité 1 pour l'interruption du timer 0 (TMR0) :

```
*(volatile unsigned long*)(ICR1) = 0x00009000;
```

Exemple en C de configuration du niveau de priorité 6 pour l'interruption du timer 0 et désactivation des interruptions INT1 à INT4 et TMR1 à TMR3 :

```
*(volatile unsigned long*)(ICR1) = 0x8888E888;
```

Utilisation du bit PI pour les sources externes ($\overline{\text{INT1}}$ à $\overline{\text{INT6}}$)

Pour chacune des 6 sources externes d'interruption, l'écriture d'un 1 sur le bit PI a une double fonction : en plus de la validation de l'écriture des bits IPL, il sert aussi à acquitter la demande d'interruption (cf. principe de l'acquiescement au §6.4.4). **Pour les autres modules, les bits d'acquiescement se trouvent dans les registres propres à chaque module (cf. bits CAP et REF des registres TERN des timers).** La remise à zéro de ces bits provoque automatiquement la remise à zéro des bits PI correspondant. En revanche, la remise à zéro du bit PI n'a pas d'effet sur le flag du module correspondant.

Lors de l'acquiescement d'une interruption externe, il est nécessaire de repositionner le niveau IPL de l'interruption concernée, puisque l'écriture d'un 1 sur PI va provoquer sa modification. Si on ne connaît pas IPL, on effectuera un masquage afin de ne pas le modifier.

Exemple en C d'acquittement de l'interruption externe $\overline{\text{INT2}}$ (priorité = 2) :

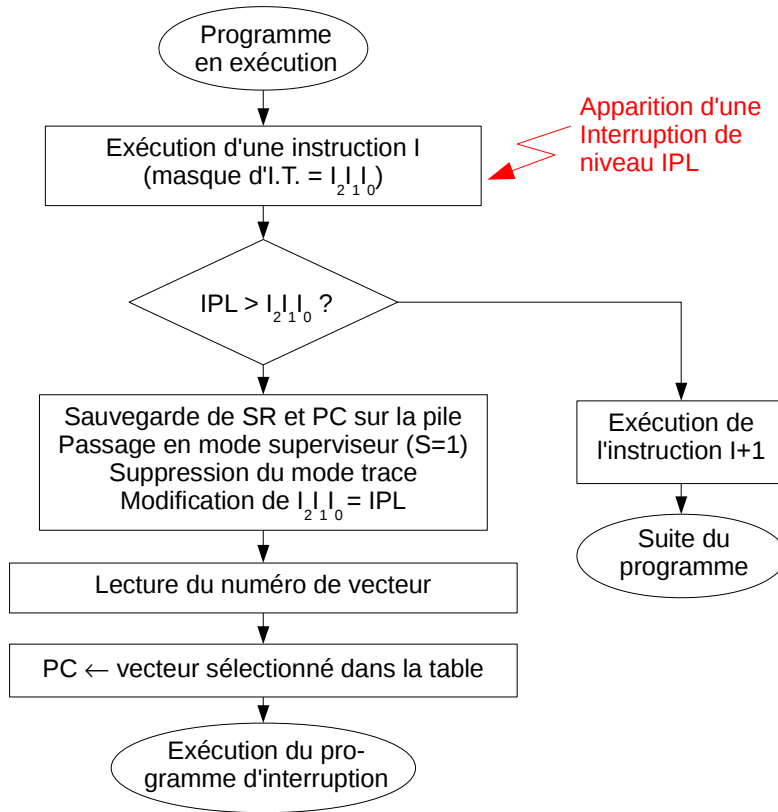
```
*(volatile unsigned long*)(ICR1) = 0x0A000000;
```

Exemple en C d'acquittement de l'interruption externe $\overline{\text{INT2}}$ (priorité inconnue donc masquage) :

```
*(volatile unsigned long*)(ICR1) =  
    *(volatile unsigned long*)(ICR1) | 0x08000000;
```

8.3.2. Traitement d'une interruption

L'organigramme ci-dessous décrit les actions liées au traitement d'une interruption par le Coldfire :



8.4. Registres spécifiques

8.4.1. Registre de réveil (Programmable Interrupt Wakeup Register : PIWR)

Le registre de réveil configure quelles sources d'interruption sont autorisées à réveiller l'unité centrale du mode veille (SLEEP) ou du mode STOP. Seules les sources externes (INT1 à INT6) peuvent faire sortir l'unité centrale du mode STOP.

Les 32 bits du registre PIWR sont détaillés au §9, (Tableaux récapitulatifs)

Niveau logiques des bits de PIWR :

- 0 : la source correspondante n'a pas le droit de réveiller l'unité centrale
- 1 : la source correspondante peut réveiller l'unité centrale

8.5. Tables des vecteurs

8.5.1. Table des vecteurs d'interruption du MCF5272

Chaque vecteur contient une adresse (donc un mot binaire 32 bits, stocké dans 4 cases mémoire). On a donc la relation suivante entre les numéros de vecteurs et leurs adresses :

$$\text{adresse vecteur} = 4 \times \text{n}^\circ \text{ de vecteur}$$

Numéro de vecteur	Adresse du vecteur (hexa)	Source matérielle	Fonction matérielle
64	0x100	-	Interruption parasite
65	0x104	INT1	Interruption externe n°1
66	0x108	INT2	Interruption externe n°2
67	0x10C	INT3	Interruption externe n°3
68	0x110	INT4	Interruption externe n°4
69	0x114	TMR0	Timer 0
70	0x118	TMR1	Timer 1
71	0x11C	TMR2	Timer 2
72	0x120	TMR3	Timer 3
73	0x124	UART1	Liaison série 1
74	0x128	UART2	Liaison série 2
75	0x12C	PLIP	PLIC 2kHz périodique
76	0x130	PLIA	PLIC asynchrone
77	0x134	USB0	Terminaison USB 0
78	0x138	USB1	Terminaison USB 1
79	0x13C	USB2	Terminaison USB 2
80	0x140	USB3	Terminaison USB 3
81	0x144	USB4	Terminaison USB 4
82	0x148	USB5	Terminaison USB 5
83	0x14C	USB6	Terminaison USB 6
84	0x150	USB7	Terminaison USB 7
85	0x154	DMA	Contrôleur de DMA
86	0x158	ERx	Ethernet (récepteur)
87	0x15C	ETx	Ethernet (émetteur))
88	0x160	ENTC	Ethernet (module NTC)
89	0x164	QSPI	Interface série QSPI
90	0x168	INT5	Interruption externe n°5
91	0x16C	INT6	Interruption externe n°6 (B.P. Abort sur kit Cobra)
92	0x170	SWT0	Débordement timer chien de garde
93	0x174	-	inutilisé
94	0x178	-	inutilisé
95	0x17C	-	inutilisé

8.5.2. Table des vecteurs d'exception

Chaque vecteur contient une adresse (donc un mot binaire 32 bits stocké dans 4 cases mémoire). On a donc la relation suivante entre les numéros de vecteurs et leurs adresses :

$$\text{adresse vecteur} = 4 \times \text{n}^\circ \text{ de vecteur}$$

Numéro de vecteur	Adresse du vecteur (hexa)	Fonction logicielle
0	0x00	Pointeur de pile au reset
1	0x04	PC au reset
2	0x08	Erreur bus
3	0x0C	Erreur adresse
4	0x10	Instruction illégale
5	0x14	Division par zéro
6-7	0x18-0x1C	inutilisé
8	0x20	Violation de privilège
9	0x24	Mode trace
10	0x28	Instruction illégale (line-a)
11	0x2C	Instruction illégale (line-f)
12	0x30	Point d'arrêt
13	0x34	inutilisé
14	0x38	Erreur de format
15	0x3C	Interruption non initialisée
16-23	0x40-0x5C	inutilisé
24	0x60	Interruption parasite
25-31	0x64-0x7C	inutilisé
32-47	0x80-0xBC	Instructions Trap #0-15
48-55	0xC0-0xDC	Exceptions unité virgule flottante
56-60	0xE0-0xF0	inutilisé
61	0xF4	Instruction non supportée
62-63	0xF8-0xFC	inutilisé
64-255	0x100-0x3FC	Espace réservé pour la table des vecteurs d'interruption

9. Tableaux récapitulatifs des registres des modules périphériques et du contrôleur d'interruptions

9.1. Registres du contrôleur d'interruptions

Registres de configuration ICR1 à ICR4 :

Reg.	Adresse	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR1	0x10000020	INT1 PI	INT1IPL			INT2 PI	INT2IPL			INT3 PI	INT3IPL			INT4 PI	INT4IPL			TMR0 PI	TMR0IPL			TMR1 PI	TMR1IPL			TMR2 PI	TMR2IPL			TMR3 PI	TMR3IPL		
ICR2	0x10000024	UART 1 PI	UART1IPL			UART 2 PI	UART2IPL			PLIP PI	PLIPIPL			PLIA PI	PLIAIPL			USB0 PI	USB0IPL			USB1 PI	USB1IPL			USB2 PI	USB2IPL			USB3 PI	USB3IPL		
ICR3	0x10000028	USB4 PI	USB4IPL			USB5 PI	USB5IPL			USB6 PI	USB6IPL			USB7 PI	USB7IPL			DMA PI	DLMAIPL			ERx PI	ERxIPL			ETx PI	ETxIPL			ENTC PI	ENTCIPL		
ICR4	0x1000002C	QSPI PI	QSPIIPL			INT5 PI	INT5IPL			INT6 PI	INT6IPL			SWT0 PI	SWTOIPL			-	-			-	-			-	-			-	-		

Bits PI : en lecture, drapeau d'interruption en attente de traitement. En écriture, **un '1' acquitte la demande d'interruption**, c'est à dire remet à '0' le bit PI.

Bits IPL : niveau de priorité. 7=interruption la plus prioritaire, 0=interruption désactivée. L'écriture de IPL se fait **en écrivant simultanément un '1' sur le bit PI**.

Registre de sélection de front actif PITR :

Reg.	Adresse	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PITR	0x10000034	INT1	INT2	INT3	INT4	-											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-										INT5	INT6	-			

Sélection du front actif pour chaque source externe d'interruption : '1' front montant, '0' front descendant

Registre d'état des sources ISR :

Reg.	Adresse	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISR	0x10000030	INT1	INT2	INT3	INT4	TMR0	TMR1	TMR2	TMR3	UART1	UART2	PLIP	PLIA	USB0	USB1	USB2	USB3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB4	USB5	USB6	USB7	DMA	ERx	ETx	ENTC	QSPI	INT5	INT6	SWTO	-	-	-	-

Registre de réveil PIWR :

Reg.	Adresse	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIWR	0x10000038	INT1	INT2	INT3	INT4	TMR0	TMR1	TMR2	TMR3	UART1	UART2	PLIP	PLIA	USB0	USB1	USB2	USB3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB4	USB5	USB6	USB7	DMA	ERx	ETx	ENTC	QSPI	INT5	INT6	SWTO	-	-	-	-

'1' : source autorisée à réveiller l'unité centrale, '0' : pas de réveil par cette source d'interruption.

9.2. Registres des modules PWM

Registre	Adresse	Bit 7	6	5	4	3	2	1	Bit 0
PWCR0	0x100000C0	EN - Enable 0 : stop 1 : run	FCR1-Force out 0 : normal 1 : sortie à 1	LVL Niveau de sortie quand EN=0	-	CKSL – Clock select Prédiviseur : rapport = 2 ^{CKSL}			
PWCR1	0x100000C4								
PWCR2	0x100000C8								
PWWD0	0x100000D0	Largeur d'impulsion entre 0 et 255							
PWWD1	0x100000D4								
PWWD2	0x100000D8								

Registres GPIO influençant le fonctionnement (broches multi-fonctions) des modules PWM 1 et 2 : PDCNT(15..12).

9.3. Registres des timers

Registre	Adresse	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
TMR0	0x10000200	PS - Prescaler : rapport de division d'horloge 66 Mhz= PS+1								CE – Capture Edge : validation de la capture et front actif	OM	ORI Output Reference Interrupt enable	FRR 0:FreeRun 1:Restart	CLK 00 : arrêté 01 : 66MHz 10 : 4,125MHz 11 : Tin (TMR0-1)		RST 0 : stop 1 : 0 reset 1 : run	
TMR1	0x10000220										0 impulsion 1 inversion						
TMR2	0x10000240										-						
TMR3	0x10000260																
TRR0	0x10000204	Référence 16 bits comparée à TCN															
TRR1	0x10000224																
TRR2	0x10000244																
TRR3	0x10000264																
TCAP0	0x10000208	Capture 16 bits du compteur suite à un évènement															
TCAP1	0x10000228																
TCAP2	0x10000248																
TCAP3	0x10000268																
TCN0	0x1000020C	Compteur de timer 16 bits – une écriture provoque sa remise à 0															
TCN1	0x1000022C																
TCN2	0x1000024C																
TCN3	0x1000026C																
TER0	0x10000210	-														REF evenem ^t TCN=TRR survenu	CAP evenem ^t capture survenu
TER1	0x10000230																
TER2	0x10000250																
TER3	0x10000260																

Registres GPIO influençant le fonctionnement (broches multi-fonctions) des timers : PBCNT et PDCNT.

9.4. Registres de configuration des broches multifonctions

Registre	Adresse		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PACNT	0x10000080	00 01 10	PA15 DGNT1 -		PA14 DREQ1 -		PA13 DFC3 -		PA12 DFSC2 -		PA11 - QSPI_CS1		PA10 DREQ0 -		PA9 DGNT0 -		PA8 FSC0/FSR0 -		PA7 QSPI_CS3 DOUT3		PA6 USB_RxD -		PA5 USB_TxEN -		PA4 USB_susp -		PA3 USB_TN -		PA2 USB_RN -		PA1 USB_RP -		PA0 USB_TP -	
PBCNT	0x10000088	00 01	PB15 EMDC		PB14 ETH		PB13 ETH		PB12 ETH		PB11 ETH		PB10 ETH		PB9 ETH		PB8 ETH		PB7 TOUT0		PB6 -		PB5 TA		PB4 URT0_CLK		PB3 URT0_RTS		PB2 URT0_CTS		PB1 URT0_RxD		PB0 URT0_TxD	
PDCNT	0x10000098	00 01 10 11	Inutilisé, laisser à 0																HZ PWM_OUT2 TIN1 -	HZ PWM_OUT1 TOUT1 -	HZ - DIN3 INT4	HZ DOUT0 URT1_TxD -	HZ - URT1_RTS INT5	HZ - URT1_CTS QSPI_CS2	HZ DIN1 URT1_RxD -	HZ DCL0 URT1_CLK -								

Note :

Bien que seuls les 16 bits de poids faible de PDCNT soient utilisés, le registre doit impérativement être lu ou écrit sur 32 bits.

9.5. Registres des modules GPIO

Registre	Adresse	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
PADAT	0x10000086	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PBDAT	0x1000008E	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PCDAT	0x10000096	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PADDR	0x10000084	Chaque bit définit le sens du bit correspondant de PxDAT 1 : bit correspondant en sortie 0 : bit correspondant en entrée															
PBDDR	0x1000008C																
PCDDR	0x10000094																

Notes :

- ◆ Les broches TIN0, PWMOUT0, INT1, INT2, INT3, QSPI_CS0 ne sont pas partagées avec d'autres fonctions et ne figurent donc pas dans ce tableau. Elles ne nécessitent aucune configuration particulière.
- ◆ PA15 / DGNT1 est aussi utilisée pour l'entrée d'interruption INT6 utilisée pour la fonction Abort. L'utilisation de PA15 en GPIO désactivera de fait INT6 et le bouton Abort.
- ◆ Les cases grisées sur les ports A et B correspondent aux entrées – sorties pratiquement utilisables sur le kit Cobra. La modification de la configuration des autres broches entraînera le plantage des liaisons série/ethernet.