Floating Point Unit 32/64 bits IEEE-754

Floating Point Unit (FPU)

L'unité de calcul à virgule flottante FPU intégrable aux processeurs 32/64 bits de la famille JAP permet une exécution performante des opérations flottantes simple et double précision. Cette unité est compatible avec le standard ANSI/IEEE-754. Elle incorpore en plus les particularités liées à l'utilisation de JAVA (1), et a été élargie au calcul de la multiplication, la division et le modulo en entier simple et double précision.

La FPU a été conçue dans l'esprit de répondre aux besoins de JAVA en terme de calcul sur des nombres à virgule flottante. A cet égard, elle permet d'exécuter l'ensemble des bytes-codes JAVA relatifs aux opérations flottantes en simple et double précision ainsi que les natives de la classe java/lang/Math.java recoupant le standard IEEE-754 (cf tableau ci-dessous). Ce premier jeu d'opérations à été enrichi d'opérations plus spécifiques à la réalisation de fonctions natives en langage C-ANSI reprenant notamment les tests, les conversions, la gestion des modes d'arrondi et la gestion des exceptions tel que décrits dans la norme ANSI/IEEE-754 (cf tableau ci-dessous). A terme, les opérations disponibles seront étendues aux autres natives de la classe java/lang/Math.java comme les fonctions trigonométriques élémentaires ou le logarithme et l'exponentielle.

La FPU se connecte directement au cache pile des processeurs JAP permettant, ainsi, d'exécuter jusqu'à une opération flottante ou entière complexe (multiplication, division, modulo) à chaque cycle d'horloge sans effectuer aucun déplacement mémoire des opérandes. Le résultat de l'opération est, ensuite, renvoyé dans le cache pile. La FPU est indépendante du chemin de données de l'unité arithmétique entière et autorise une exécution concurrente des instructions flottantes et entières.

(1) Spécificités JAVA : L'arithmétique des nombres à virgule flottante utilisée dans JAVA défini un sous-ensemble du standard IEEE-754. Ce sous-ensemble suppose entre autre que toutes les exceptions sont masquées, force le mode d'arrondi « au plus près », redéfini les opérations de tests, défini le modulo comme étant équivalente à la fonction fmod en C-ANSI (drem initialement), force le mode d'arrondi « vers zéro » pour les conversions flottants vers entiers (cast en C-ANSI) et ne prend pas en compte les formats flottants étendus.

Architecture de la FPU

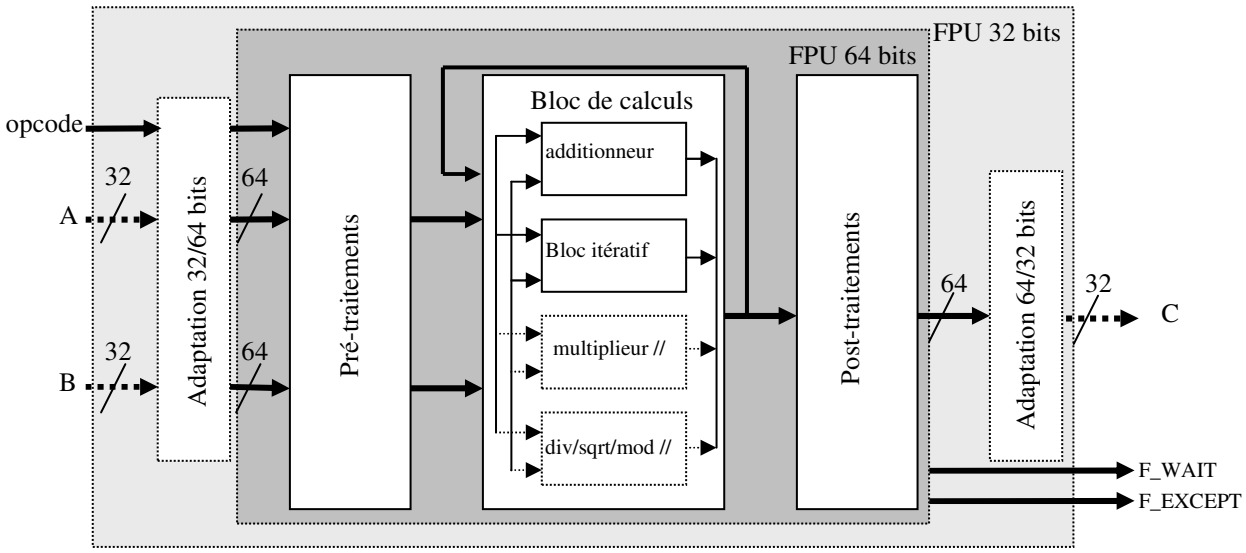
L'architecture de la FPU s'articule sur un chemin de donnée 64 bits découpé en 3 étages de pipeline comme l'illustre la figure suivante. Le cœur de calcul est basé sur un bloc addition traitant toutes les opérations élémentaires (addition, soustraction, négation, test et conversions), un bloc itératif pour les opérations de multiplication, division, modulo et racine carrée, et sur option d'opérateurs parallèles se substituant à l'exécution itérative d'une opération donnée. Cette dernière possibilité permet d'ajuster le compromis temps d'exécution/coût surface/consommation lié aux opérateurs complexes (multiplieur/diviseur/…). A terme, les opérations effectuées à partir du bloc itératif de la FPU seront étendues aux diverses natives de la classe java/lang/Math.java comme les fonctions trigonométriques élémentaires, le logarithme ou l'exponentielle. Le cœur de calcul est interne au deuxième étage du pipeline.

Le premier et le troisième étages assurent les conversions entre les diverses formats en entrée/sortie, flottants et entiers en simple et double précision, et le format de représentation interne. Ces 2 étages intègrent respectivement plusieurs pré-traitements utiles au bon fonctionnement des calculs itératifs pour le premier, et des post-traitements nécessaires à l'adéquation avec la norme ANSI/IEEE-754 et JAVA pour le second.

La FPU s'intégrant à la fois dans les processeurs 32 et 64 bits de la famille JAP, un module permet l'adaptation du chemin de données 64 bits à un chemin de données 32 bits. Ce dernier assure le synchronisme entre les entrées/sorties des données et le bon déroulement des calculs.

L'architecture de la FPU comprend un registre status de 32 bits (cf tableau ci-dessous) accessible en écriture et en lecture. Ce registre sert à positionner les modes d'arrondi et le masque d'exception, et à récupérer les exceptions comme stipulé dans la norme ANSI/IEEE-754 .

En plus du résultat des calculs, la FPU délivre un signal F_EXCEPT actif quand une exception non masquée se produit, et un signal F_WAIT signalant qu'un calcul est dans le bloc d'itération utile pour le synchronisme des données en sortie.



Performances de la FPU

Les performances indiquées ci-dessous correspondent à une version 64 bits de l'unité de calcul à virgule flottante FPU intégrant un multiplieur parallèle. Les calculs de la division, du modulo et de la racine carrée sont réalisés dans le bloc itératif.

Opérations(s)	Nb de cycles	Latence pipeline	Opérations
Byte-code JAVA compatible IEEE-754			
fadd, fsub, fmul dadd, dsub, dmul fneg, dneg	1	+2	addition, soustraction, négation et multiplication en float et double
i2f, i2d, l2f, l2d f2i, f2l, d2i, d2l f2d, d2f	1	+2	conversions int, long <=> float, double float <=> double
Fdiv	1 à 27	+2	division float
Ddiv	1 à 56	+2	division double
Frem	1 à 276	+2	modulo float
Drem	1 à 4097	+2	modulo double
Natives de math.java compatible IEEE-754			
ceil (double=>double)	1	+2	arrondi entier vers + l'infini
floor (double=>double)	1	+2	arrondi entier vers – l'infini
rint (double=>double)	1	+2	arrondi entier au plus près
sqrt (double)	1 à 56	+2	racine carrée en double
IEEEremainder (double)	1 à 2097	+2	reste division en double

Byte code JAVA spécifiques			
fcmpg, fcmpl dcmpg, dcmpl	1	+2	java test en double et float
Opérations IEEE-754 spécifiques			
sf2i, sf2l, sd2i, sd2l	1	+2	float, double => int, long
ceilf (float=>float)	1	+2	arrondi entier vers + l'infini
floorf (float=>float)	1	+2	arrondi entier vers – l'infini
rintf (float=>float)	1	+2	arrondi entier au plus près
sqrtf (float)	1 à 27	+2	racine carrée en float
fcmpa, fcmpb dcmpa, dcmpb	1	+2	tests IEEE-754 en double et float
read status write status	1 1	+2 +0	écriture/lecture de l'opération venant d'être effectuée gestion des exceptions
Byte-code JAVA entiers			
imul, lmul	1	+2	multiplication int et long
idiv, irem	1 à 32	+2	division et modulo int
ldiv, lrem	1 à 64	+2	division et modulo long

Exceptions et modes d'arrondi

La FPU intègre la gestion des exceptions et des modes d'arrondi telle que définie dans la norme ANSI/IEEE-754. La liste des exceptions prises en compte à été élargie à la signalisation de la division par zéro en entier (division et modulo).

Cette gestion prend la forme d'un registre 32 bits regroupant les indicateurs nécessaires au déroulement des opérations et les indicateurs d'état d'un calcul. Ce registre est mis à jour à la fin de chaque opération flottante. Il peut aussi être lu et écrit par l'utilisateur.

Bit	Nom	Descriptif
		Status avant masquage des exceptions
31	SST_NANQ	NaN quiet (Invalid)
30	SST_INF	inf (+/- composed with the sign)
29	SST_ZERO	null
28	SST_SIGN	minus 1 (cmpg/cmpl)
27	SST_MS_1	plus 1 (cmpg/cmpl)
26	SST_PS_1	saturation
25	SST_SAT	sign
		Status avant masquage des exceptions Indicateurs accumulés (remis à zéro par l'utilisateur)
24	SST_IDV0	integer division by zero
23	SST_INEX	inexact
22	SST_UNDF	float underflow
21	SST_OVRF	float overflow
20	SST_DIV0	float division by zero
19	SST_NANS	NaN signaling /Invalid operation
18-14		Réservé fonctionnement FPU

13 12 11 10 9 8	TRP_OVRI TRP_INEX TRP_UNDF TRP_OVRF TRP_DIV0 TRP_NANS	"piège à 1" des exceptions après masquage (remis à zéro par l'utilisateur)
		Masque d'activation des exceptions
7	MSK_IDV0	int/long division by zero
6	MSK_INEX	inexact
5	MSK_UNDF	float/double underflow
4	MSK_OVRF	float/double overflow
3	MSK_FDV0	float/double division by zero
2	MSK_NANS	invalid operation
1-0	AR_1 AR_0	positionnement du type d'arrondi
	0 0	vers zéro
	0 1	vers plus l'infini
	1 0	vers moins l'infini
	1 1	au plus près

Advanced Electronic Design
3 rue de l'éperon – 77000 MELUN
Téléphone : 01 64 52 16 96
Mail : info@a-e-d.com – web : www.a-e-d.com