

Architecture des Processeurs

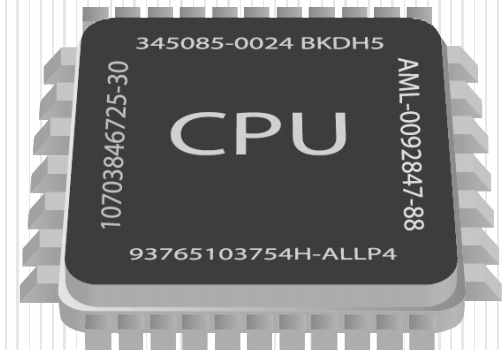
- DSP TI 66AK2H14 -

Mohammed AISSAOUI

Sebastien GARCIA

Skandar Mohamed BOULEGHEB

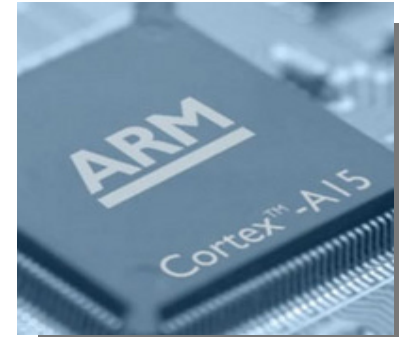
Hassouna AZOUZ



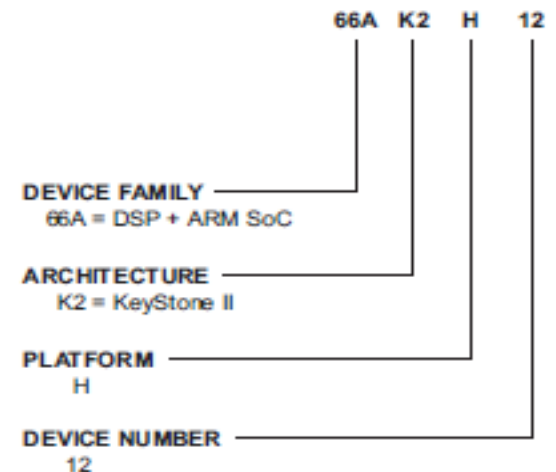
Plan

- Présentation de l'architecture
- Les éléments de performance
- La hiérarchie mémoire
- Pipeline
- Jeux d'instruction
- Instructions spécialisées
- Outils logiciels disponibles
- Domaine d'utilisation

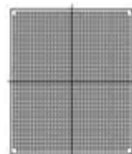
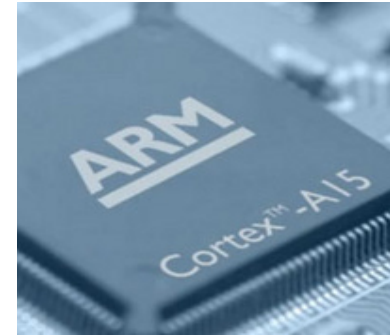
Plateforme Keystone II (Texas Instrument)



- DSP Ti 66AK2H14 (SoC):
 - Libérer de la place sur le circuit imprimé
 - Réduire le nombre de composants externes
- II Abaisser le coût de la consommation**



Plateforme Keystone II (Texas Instrument)



[Agrandir](#)

595-
66AK2H14BAAW24

66AK2H14BAAW24



Nouveau produit

Texas
Instruments

Processeurs
et
contrôleurs à
signaux
numériques -
DSP, DSC
Multicore
DSP+ARM
Keystone II
SoC

Fiche
technique

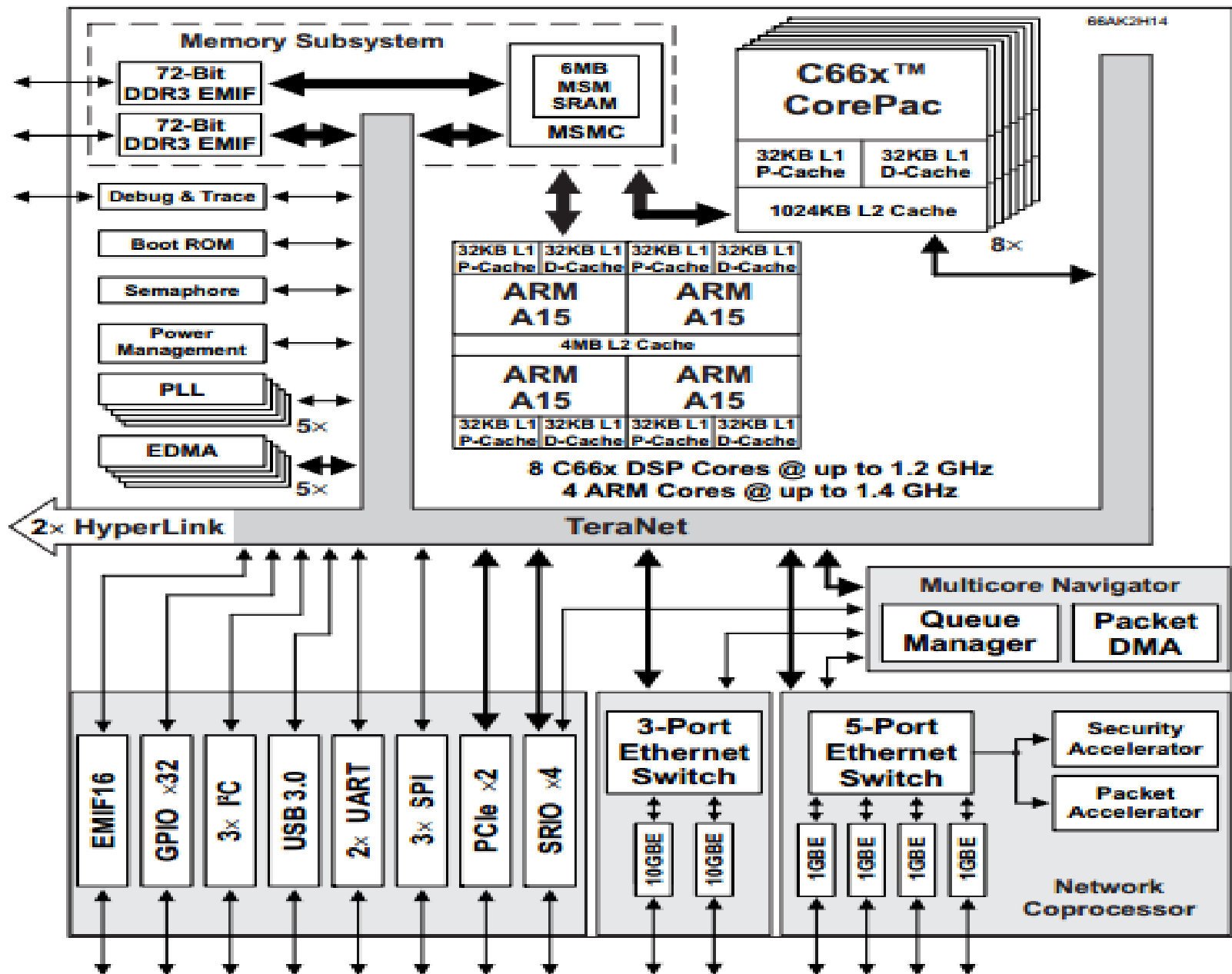
20
En stock

1: € 810,88
2: € 810,64

Les éléments de performance

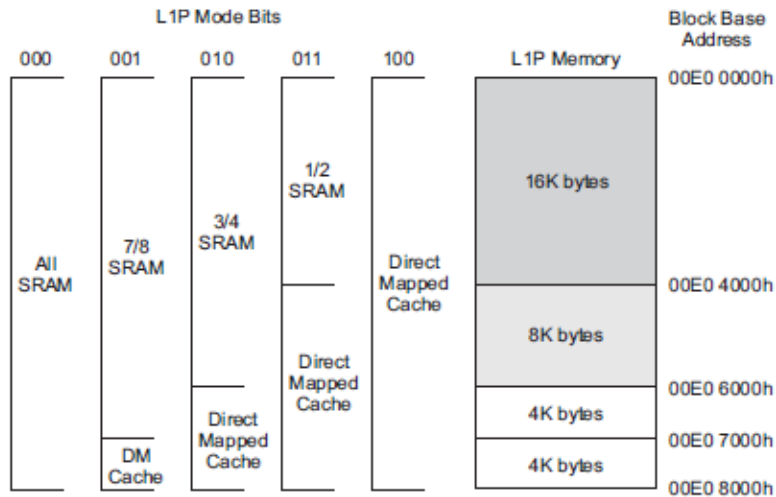
Performances	DSP TMS320C66x	ARM Cortex A-15 MPCore
Cœurs	8	4
Flops	19,2 Gflops/Cœur (153)	-
Horloge	1,2 Ghz	1,4 Ghz
Mémoire (cache)	<ul style="list-style-type: none"> - 32 K Byte L1P / L1D/Cœur - 1024 K Byte L2 / Cœur 	<ul style="list-style-type: none"> - 32 KB L1P / L1D/Cœur - 4 MB L2 pour l'ARM
Mémoire (partagée)	<ul style="list-style-type: none"> - 6 Mo MSM SRAM - Mémoire de protection 	
Voltage	- SmartReflex, tension variable (<i>DVFS</i>)	
Technologie CPU	0,028 um	

Figure 1-1 Functional Block Diagram for 66AK2H14

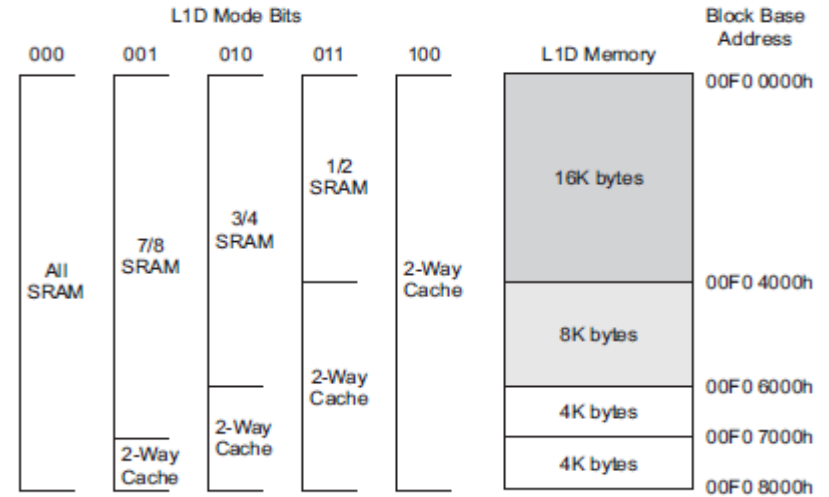


La hiérarchie mémoire

Cache:



Correspondance directe



Associative à deux sens

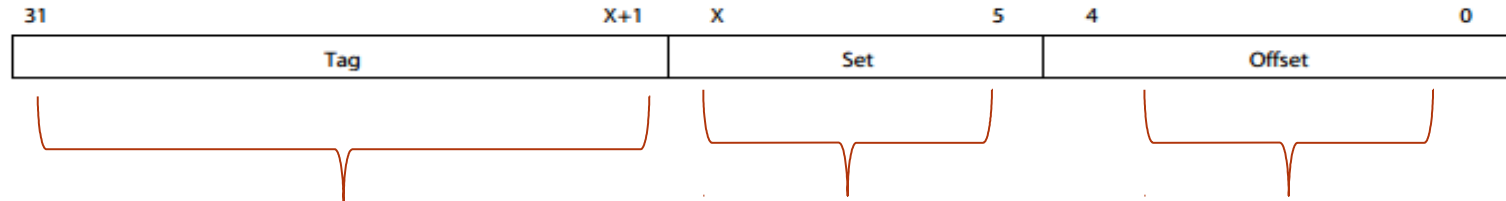
- Récupérer le code du programme à une vitesse d'horloge afin de maintenir une mémoire système étendue.

La hiérarchie mémoire

L1P (level 1 Program) :

- Une seule possibilité d'emplacement sur le cache.
- DSP tente d'extraire un morceau de code, L1P doit vérifier l'existence de l'adresse demandée

Figure 2-1 Data Access Address Organization



- Identifier l'emplacement physique réel de l'élément de données

- Adresse de la ligne de cache où les données résident

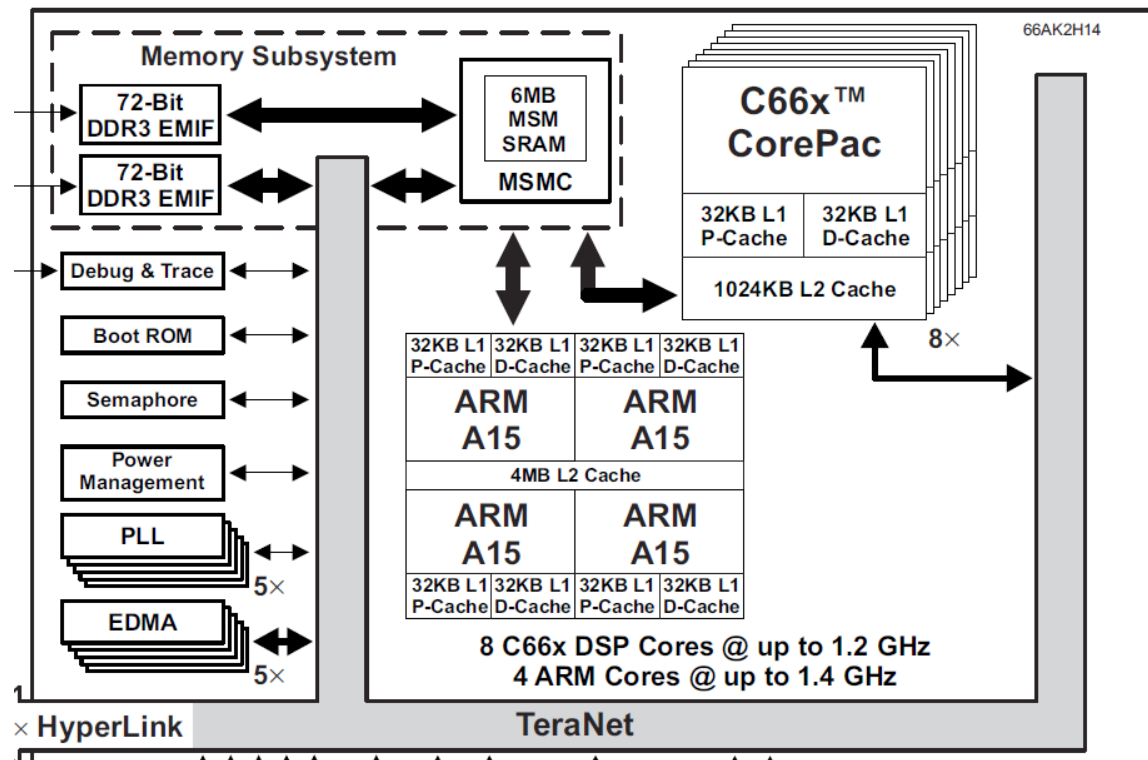
- La taille d'une ligne L1P est de 32 bytes

- Si Tag correspond et Valid = set
⇒ « hit » les données sont lues directement à partir de l'emplacement de L1P et renvoyées au

- Dépend de la quantité de L1P configurée comme cache

Protection de la mémoire

- Cache DSP réparti en pages
- Autorisation d'accès



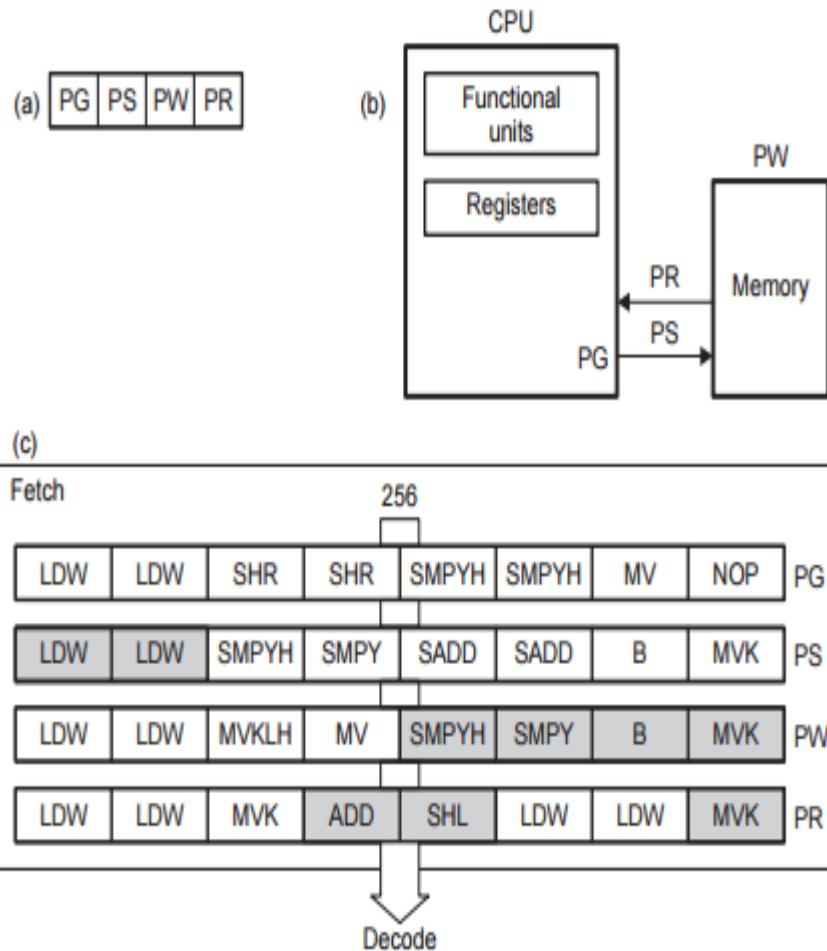
Pipeline

- 8 instructions traitées par cycle
- Instructions en parallèle ou en série



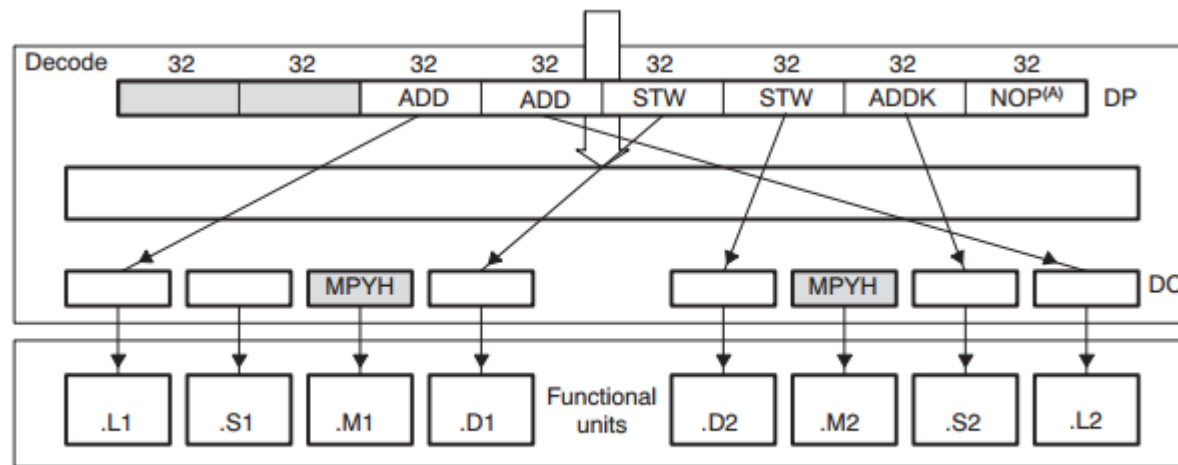
- Pipeline divisé en 3 étapes :
 - Fetch (4 sous-étapes)
 - Decode (2 sous-étapes)
 - Execute (1-5 sous-étapes)

Pipeline : Fetch



- PG: Program address generate
- PS: Program address send
- PW: Program access ready wait
- PR: Program fetch packet receive

Pipeline : Decode



(A) NOP is not dispatched to a functional unit.

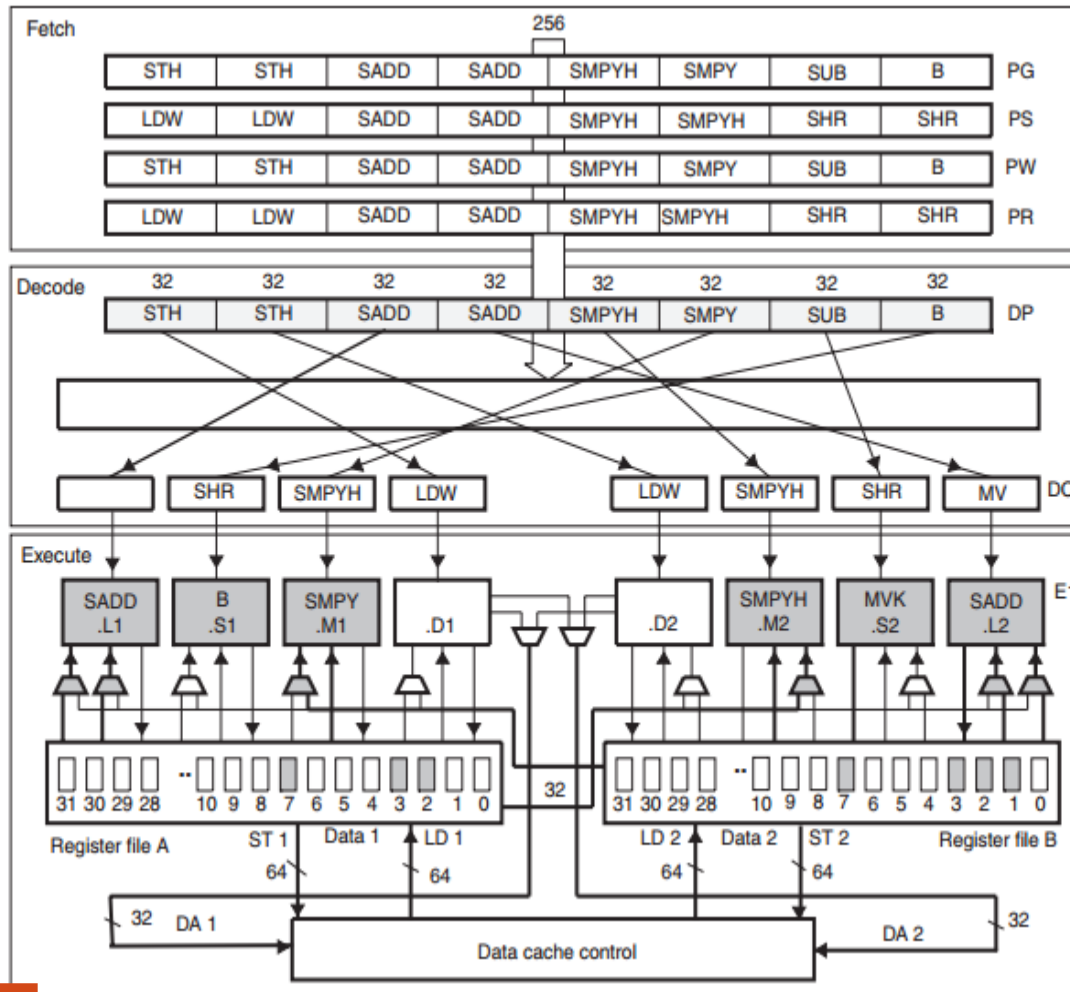
- DP: Instruction dispatch
- DC: Instruction decode

Pipeline : Execute

Execution Phase ^{1,2}	Instruction Type					
	Single Cycle	16 x 16 Single Multiply/.M Unit Nonmultiply	Store	Multiply Extensions	Load	Branch
E1	Compute result and write to register	Read operands and start computations	Compute address	Reads operands and start computations	Compute address	Target code in PG ³
E2		Compute result and write to register	Send address and data to memory		Send address to memory	
E3			Access memory		Access memory	
E4				Write results to register	Send data back to CPU	
E5					Write data into register	

- 1-5 phases
- Dépend de l'instruction

Pipeline : Exemple



SADD.L1 A2,A7,A2; E1 Phase

SADD.L2 B2,B7,B2

SMPYH.M2XB3,A3,B2

SMPY.M1XB3,A3,A2

B .S1 LOOP1

MVK .S2 117,B1

LDW .D2 *B4++,B3; DC Phase

LDW .D1 *A4++,A3

MV .L2XA1,B0

SMPYH.M1A2,A2,A0

SMPYH.M2B2,B2,B10

SHR .S1 A2,16,A5

SHR .S2 B2,16,B5

LOOP1:

STH .D1 A5,*A8++[2]; DP, PW, and PG Phases

STH .D2 B5,*B8++[2]

SADD.L1 A2,A7,A2

SADD.L2 B2,B7,B2

SMPYH.M2XB3,A3,B2

SMPY.M1XB3,A3,A2

[B1] B.S1LOOP1

[B1] SUB.S2B1,1,B1

LDW .D2 *B4++,B3: PR and PS Phases

LDW .D1 *A4++,A3

SADD.L1 A0,A1,A1

SADD.L2 B10,B0,B0

SMPYH.M1A2,A2,A0

SMPYH.M2B2,B2,B10

SHR .S1 A2,16,A5

SHR .S2 B2,16,B5


Jeux d'instruction

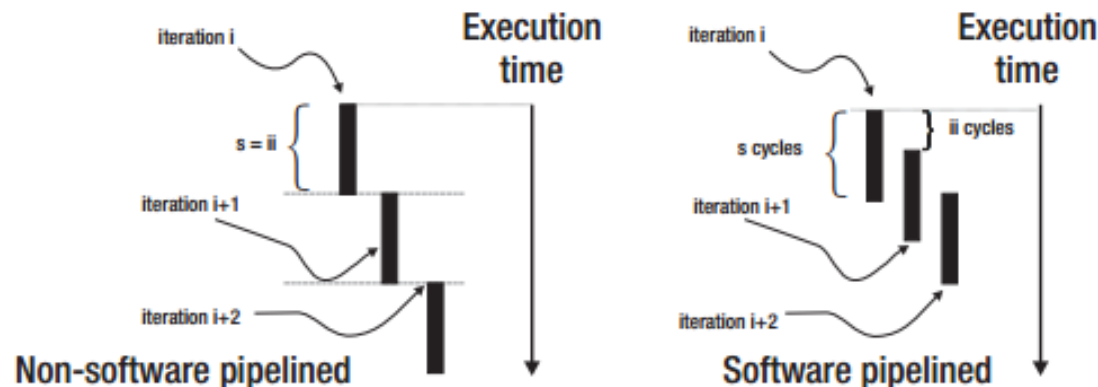
- 64 registres 32 bits, ~700 instructions
- Jusqu'à 8 instructions (codées sur 32 bit) en parallèles
- 8 unités de calcul :
 - 2 multiplieurs (1 produit 32 bit, 2 produits 16 bit ou 4 produits 8 bit par multiplieur)
 - 6 ALU
- Exécution conditionnelle des instructions
- Pipeline logicielle

Jeux d'instruction : calcul

- Données sur 8-32 bits (add, sub, mul ...)
- Normalisation et saturation
- Support Nombres complexes (64-32 bits -> 32-16 bits partie réelle, 32-16 bits partie imaginaire)
- Format virgule flottante :
 - configuration par registres : FADCR, FAUCR et FMCR
 - Simple (32 bits) et double précision (64 bits)
- Calcul vectoriel et matriciel :
 - addition, produit, conjugué, ...

Jeux d'instruction : Software Pipelined Loop

<pre>for (I=0; i<val; I++) { dest[i]=source[i]; }</pre>		MVC	8, ILC	<pre>;Do 8 loops ;4 cycle for ILC to load ;Iteration interval is 1 ;Load source ;Wait for source to load ;Position data for write ;End loop and store value</pre>
		NOP	3	
		SPLOOP	1	
		LDW	*A1++, A2	
		NOP	4	
		MV	.L1X A2, B2	
SPKERNEL	6, 0			
		STW	B2, *B0++	



Jeux d'instruction : Produit complexe vecteur 1x2 avec matrice 2x2

CMATMPY A7:A6,A3:A2:A1:A0,A11:A10:A9:A8



$$[A11:A10 \ A9:A8] = [A7 \ A6] * \begin{bmatrix} A3 & A2 \\ A1 & A0 \end{bmatrix}$$

-or-

$$\begin{aligned} A1:A10 &= A7*A3 + A6*A1 \\ A9:A8 &= A7*A2 + A6*A0 \end{aligned}$$



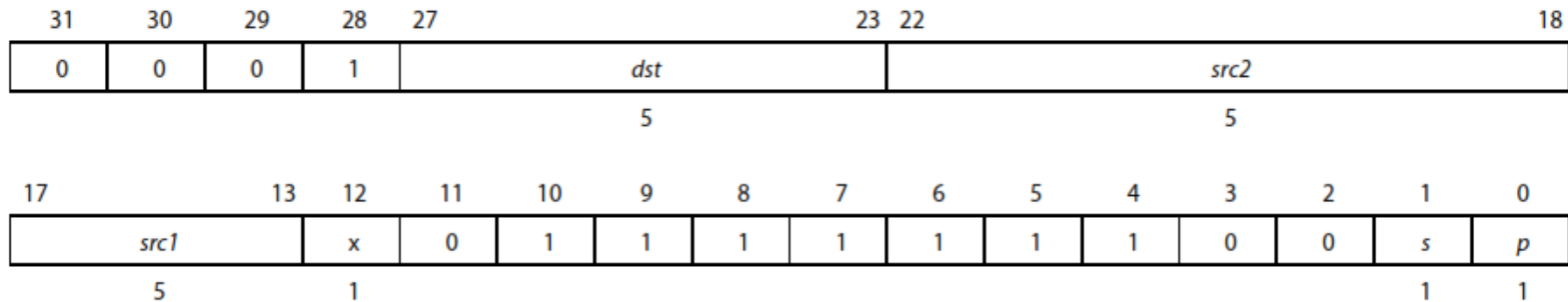
CMPY A3,A7,A31:A30
CMPY A1,A6,A29:A28
CMPY A2,A7,A27:A26
CMPY A0,A6,A25:A24
NOP
DSADD A31:A30,A29:A28,A11:A10
DSADD A27:A26,A25:A24,A9:A8

Multiplication champs de Galois

Galois Field Multiply

Syntax **GMPY** (.unit) *src1, src2, dst*

Opcode



- Théorie de Galois
- Module GMPY du DSP
- 4 cycles
- Communication : cryptographie

Outils logiciels: Développement

- Les TI 66AK2Hxx sont co-conçus par les architectes CPU, les concepteurs de compilateurs et les ingénieurs logiciels.
- C / C++
- EVMK2H: * 66AK2H14
 - * LCD
 - * Sorties Ethernet / RS232 / USB
 - * MCSDK



Outils logiciels: Développement

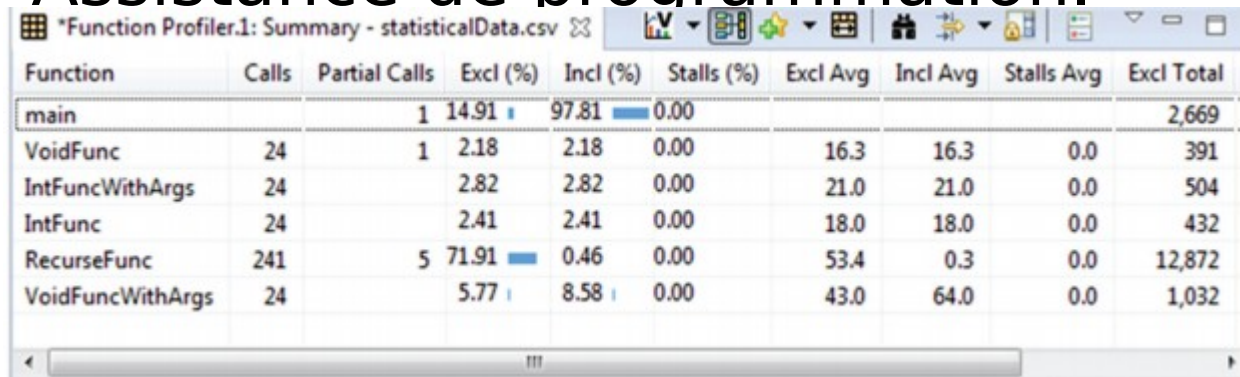
- Ordinateur: drivers pour Windows/Linux
- DSP: Code Composer Studio Integrated Development Environment (CCS)

Outils logiciels: Compilateur

- C6000 :

- * généralement utiliser en conjonction avec CCStudio pour bien profiter des performances.

- * Assistance de programmation.



The screenshot shows a window titled '*Function Profiler.1: Summary - statisticalData.csv'. It contains a table with the following data:

Function	Calls	Partial Calls	Excl (%)	Incl (%)	Stalls (%)	Excl Avg	Incl Avg	Stalls Avg	Excl Total
main		1	14.91	97.81	0.00				2,669
VoidFunc	24	1	2.18	2.18	0.00	16.3	16.3	0.0	391
IntFuncWithArgs	24		2.82	2.82	0.00	21.0	21.0	0.0	504
IntFunc	24		2.41	2.41	0.00	18.0	18.0	0.0	432
RecurseFunc	241	5	71.91	0.46	0.00	53.4	0.3	0.0	12,872
VoidFuncWithArgs	24		5.77	8.58	0.00	43.0	64.0	0.0	1,032

Outils logiciels: Librairies

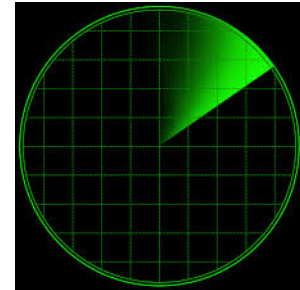
- Bibliothèques CSL (Chip Support Library)
- Bibliothèque DSPLIB (Digital Signal Processing Library)
- Bibliothèque DSPMath
- Bibliothèque IMGLIB (Image Library)

Outils logiciels: Modèles

- BSDL Model (Boundary scan description language)
- IBIS Model (I/O Buffer Information Specification)
- Power Consumption Model
- FloTherm Model

Domaine d'utilisation

- Cryptographie
- Traitement du signal / image
 - Radars
 - Militaire
 - Médical



Conclusion

Références:

- ❑ **TMS320C66x DSP** Literature Number: SPRUGW0C July 2013
CorePac user guide
- ❑ **66AK2H14/12/06 Features and Description datasheet**
- ❑ Cortex™-A15 MPCore™ **Revision: r3p2 Technical Reference Manual**
- ❑ **TMS320C66x DSP** Literature Number: SPRUGH7 **CPU and Instruction Set**