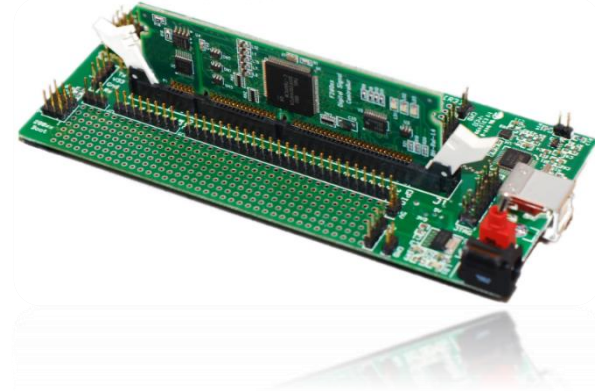
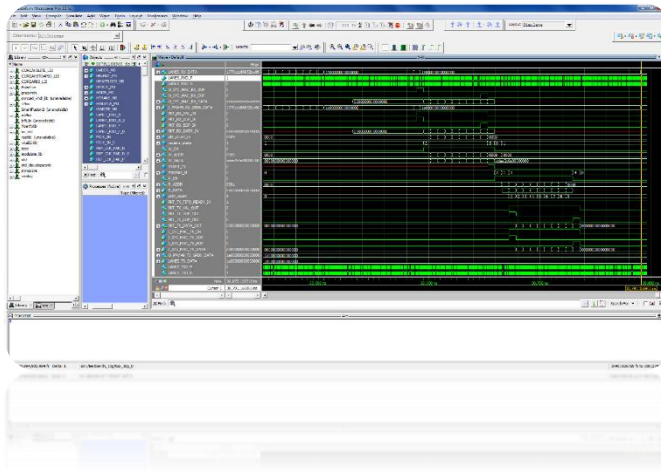
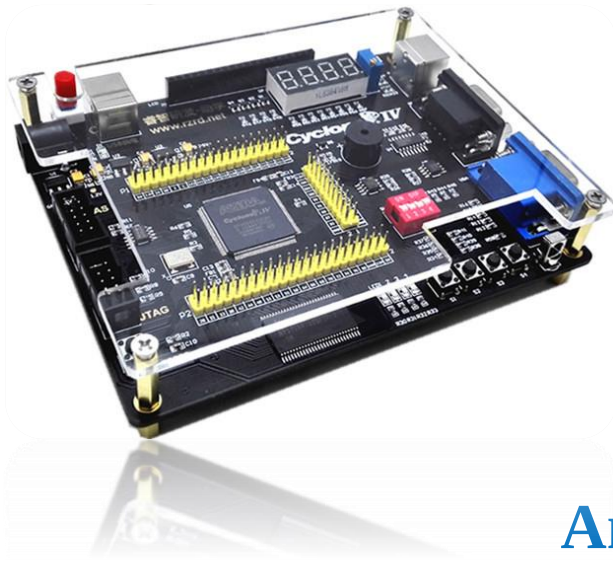


Cours FPGA, VHDL et DSP

Master d'Université Spécialisé en Ingénierie
des Systèmes Embarqués (ISE)



Année universitaire 2024/2025

Dr. Ing. LASSIOUI Abdellah

Chapitre 6

Objectifs du chapitre N6

- ✓ Comprendre la chaine de traitement d'un DSP
- ✓ Comprendre l'architecture interne d'un DSP : exemple C2000 F28335
- ✓ Comprendre les étapes principales de la programmation d'un DSP en utilisant MATLAB/Simulink

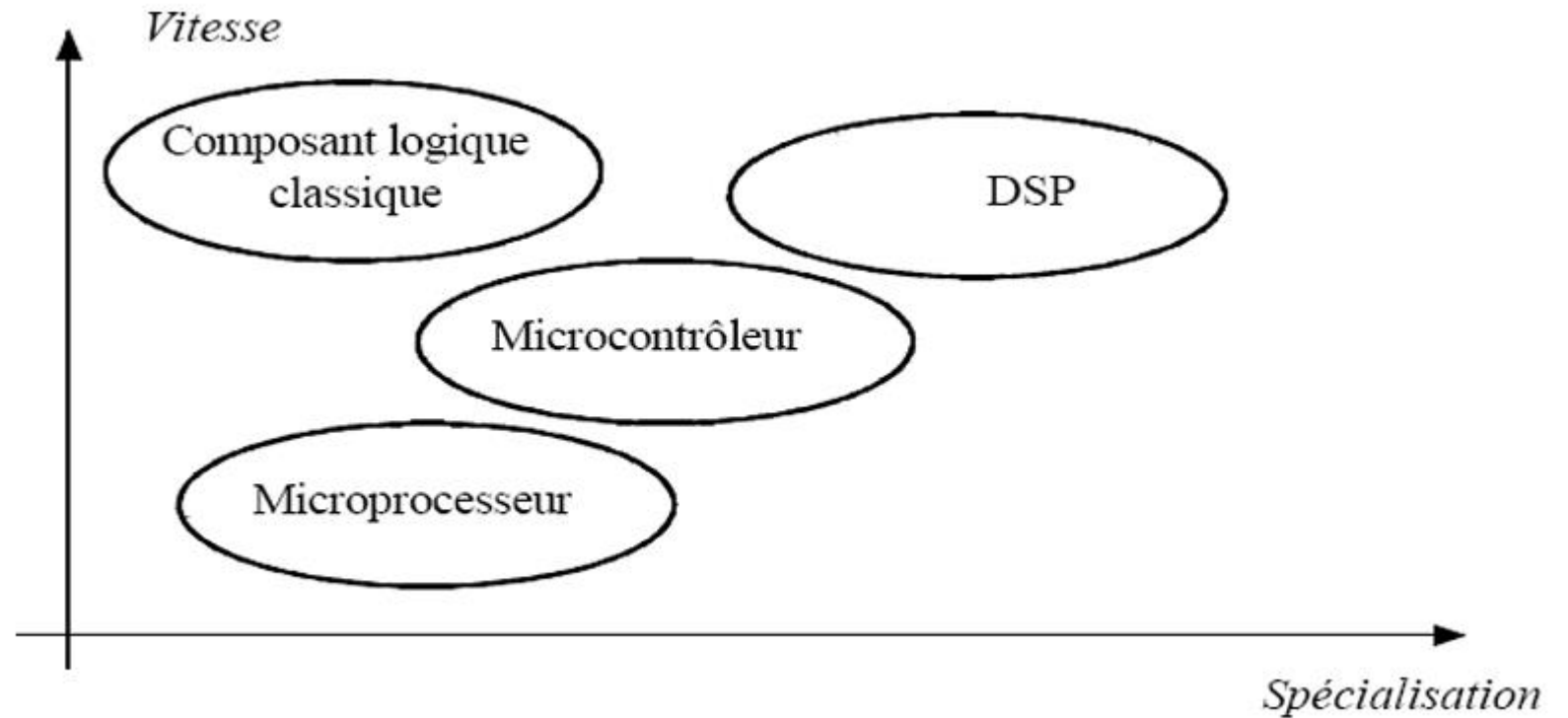
DSP: Digital Signal Processor?

- ❑ Un **DSP** est un type particulier de **microprocesseur**. Il se caractérise par le fait qu'il intègre un ensemble de fonctions spéciales. Ces fonctions sont destinées à le rendre particulièrement performant dans le domaine du **traitement numérique du signal**.

Place du DSP vis-à-vis des autres technologies

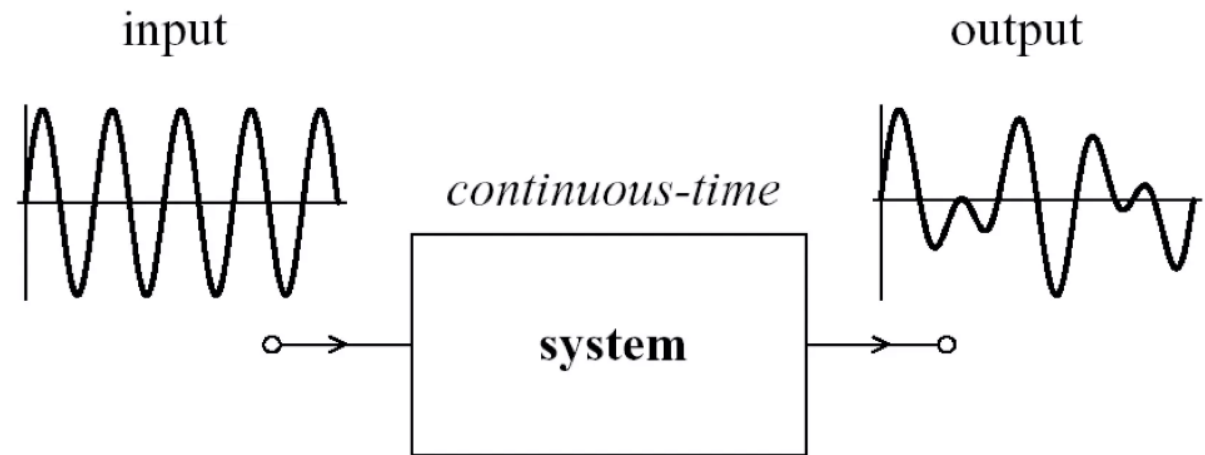
Dans la figure suivante on voit la position des DSP par rapport aux autres composants tels que les microprocesseurs et les microcontrôleurs .

Les DSP ont des vitesses d'exécutions élevées mais pour des tâches bien spécifiques



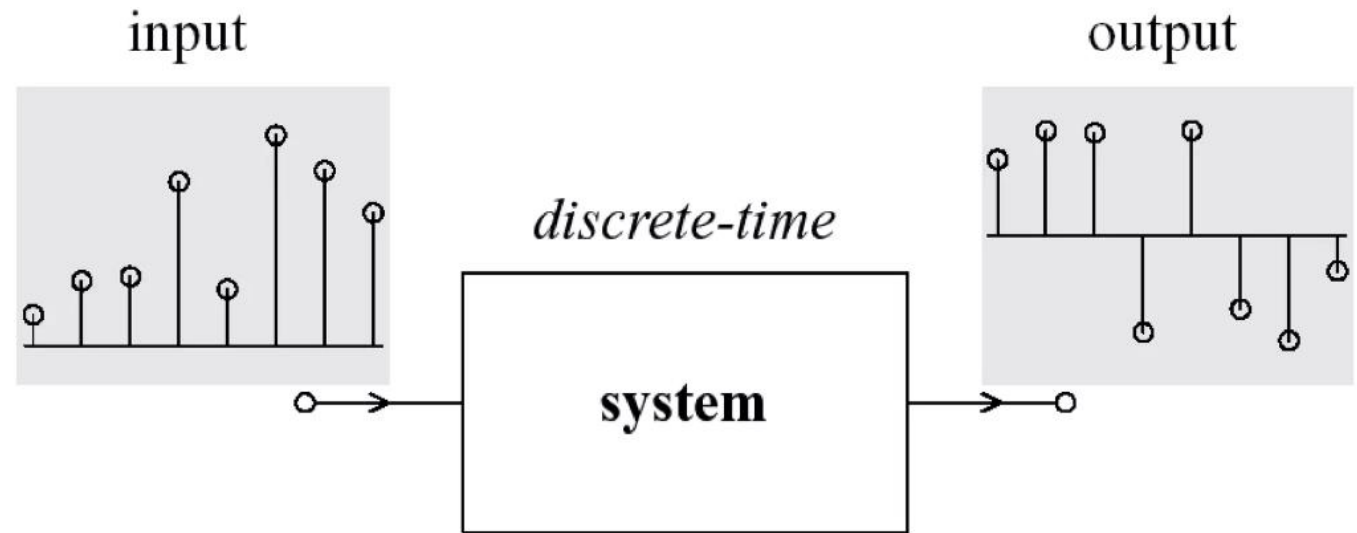
➤ **Système analogique**

- ❑ Un système avec des entrées analogiques et des sorties analogiques

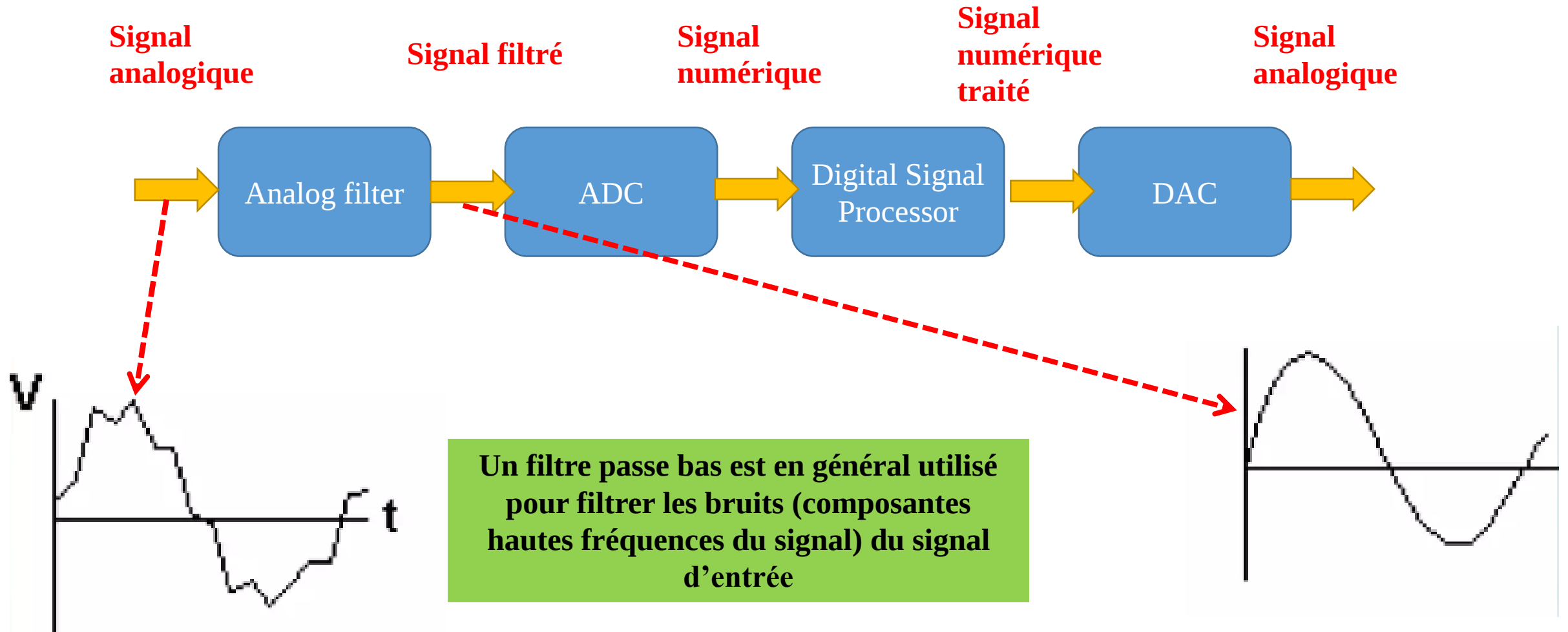


➤ **Système numérique**

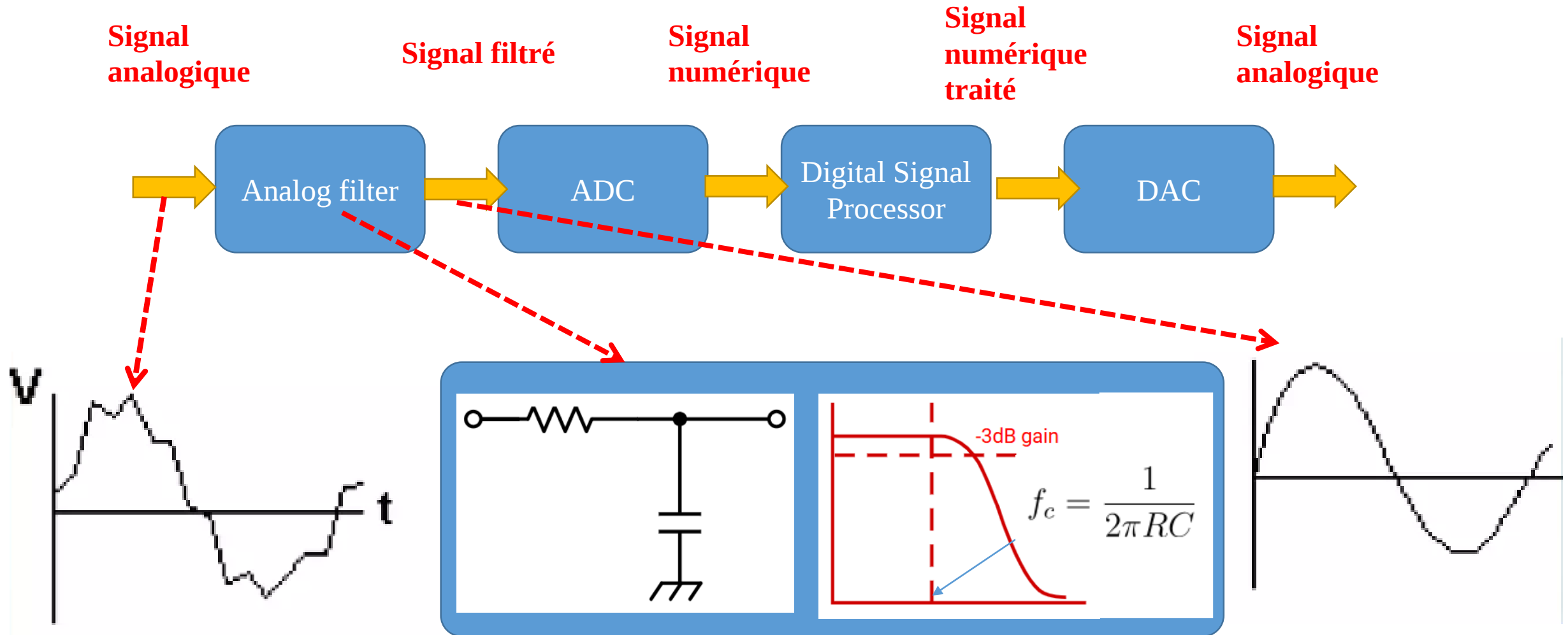
- ❑ Un système avec des entrées numériques et des sorties numériques



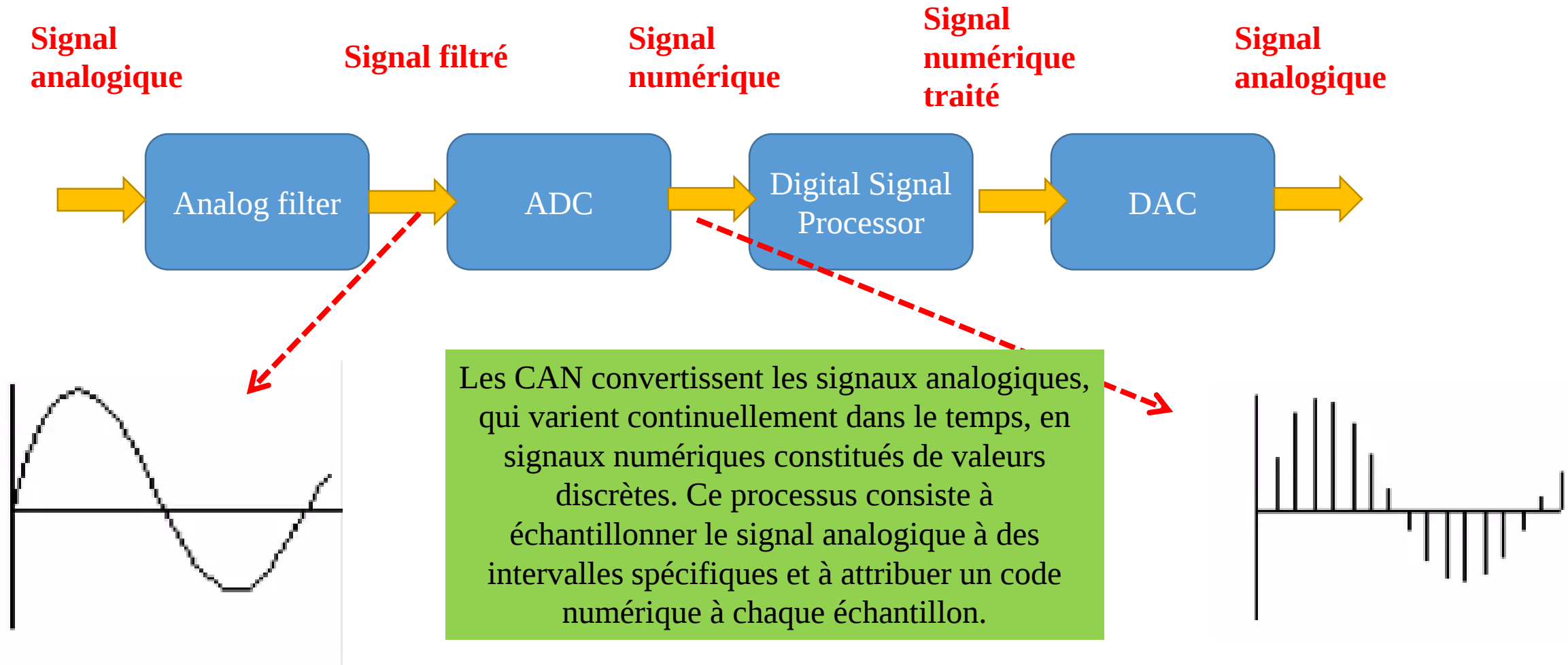
➤ Chaîne de traitement d'un DSP



➤ Chaîne de traitement d'un DSP



➤ Chaîne de traitement d'un DSP



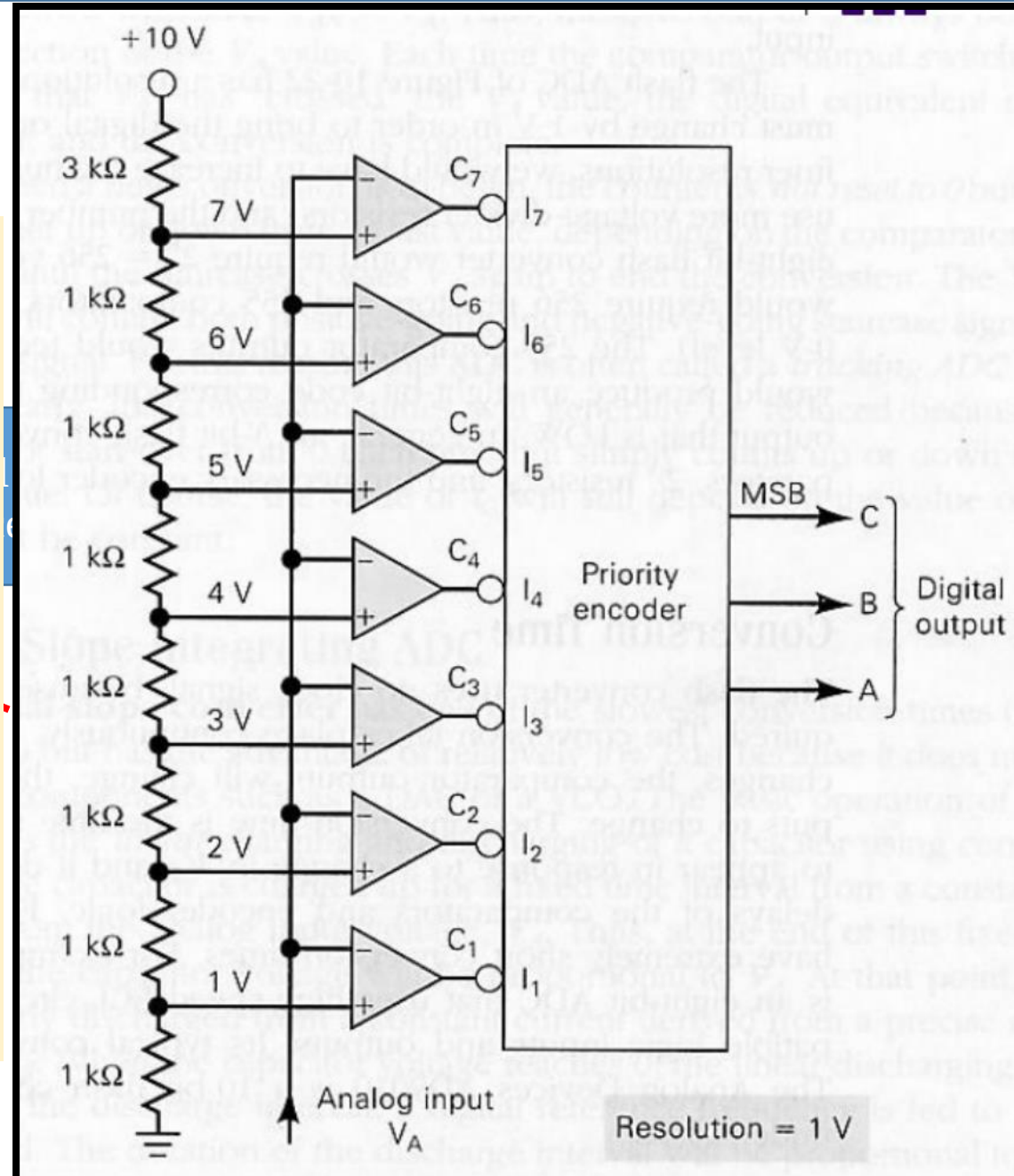
➤ Exemple d'un ADC

➤ L'ADC de type Flash

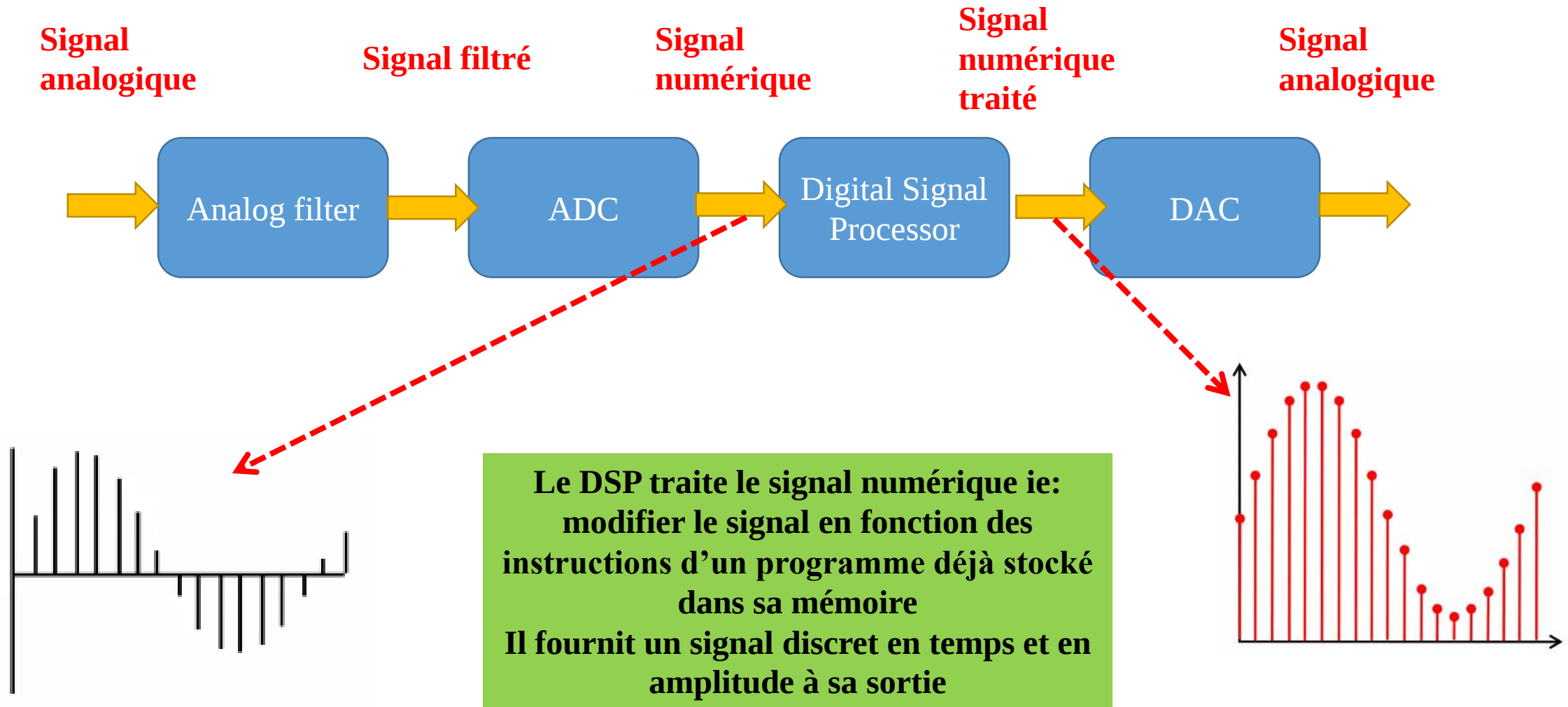
Le circuit d'un CAN de type flash est illustré dans la figure suivante:

La tension analogique est appliquée à des comparateurs dont les tensions de références sont fixées par un réseau de résistances.

Les sorties de comparateurs sont appliquées à un encodeur de priorité permettant de décoder le signal en entrée en un signal binaire sous trois bits.



➤ Chaîne de traitement d'un DSP

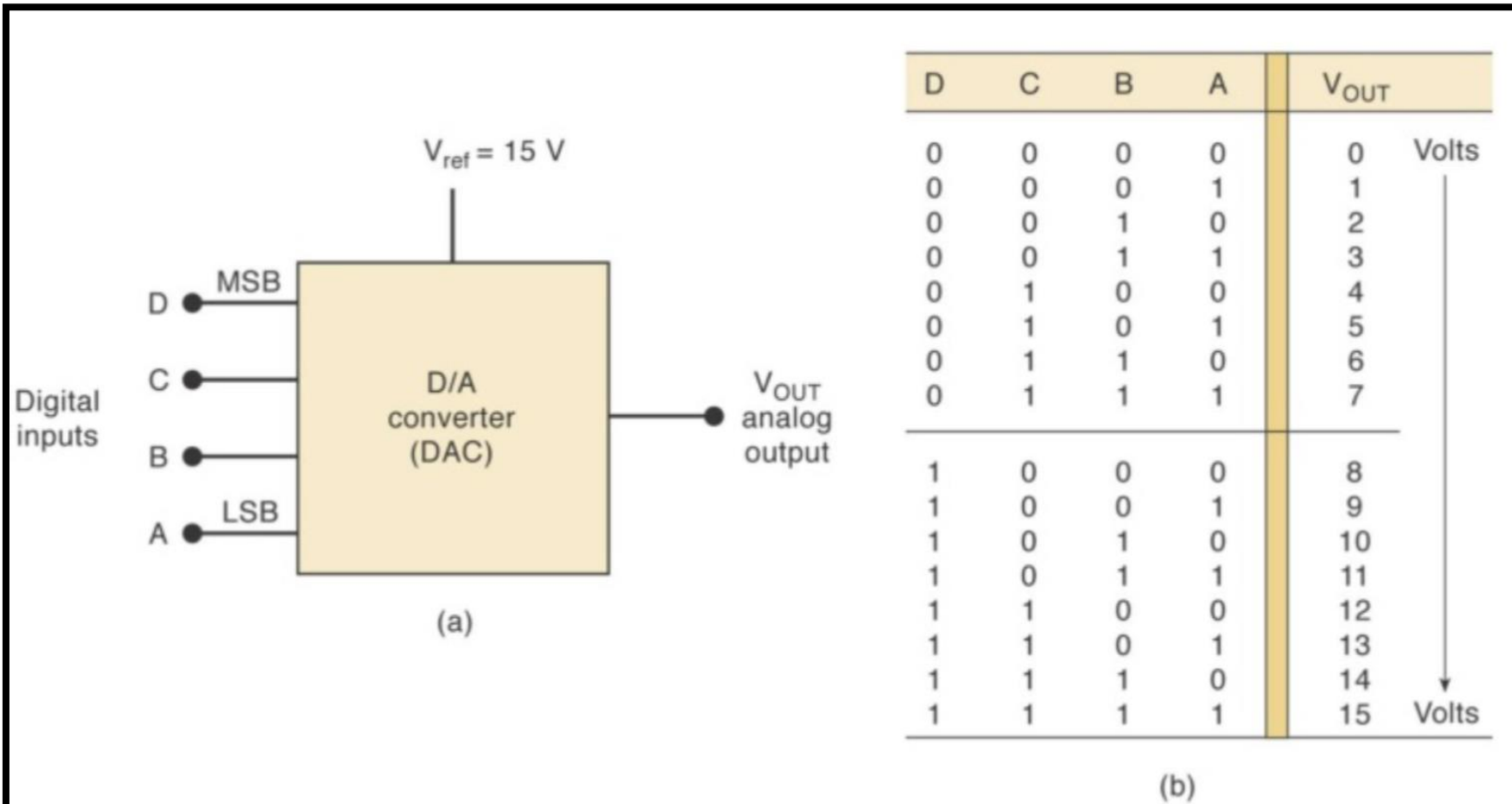


➤ Rôle d'un DAC

La figure ci-contre illustre un convertisseur numérique analogique à 4 entrées

La tension de référence V_{ref} indique la tension maximale de sortie également appelée full-scale output

La résolution dans ce cas est de 1V. Elle correspond au à la variation minimale apparaissant au niveau de la sortie pour un changement des entrées.



NB.

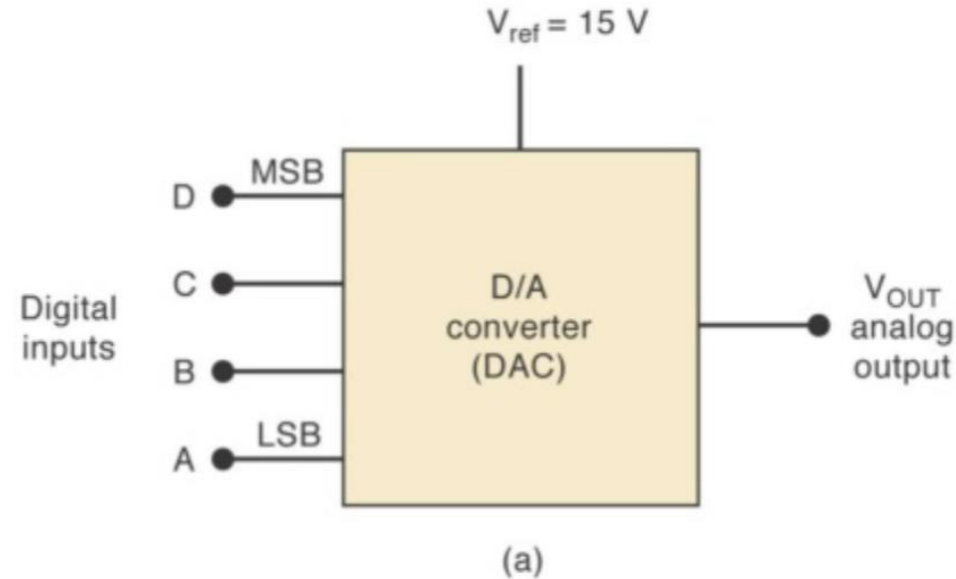
La sortie n'est pas analogique au vrai sens car elle prend seulement des valeurs Spécifiques. Cependant, si le nombre de valeurs possibles en sorties est élevée (résolution faible) la sortie à tendance à apparaître comme un signal analogique continu dans le temps.

➤ Rôle d'un DAC

La figure ci-contre illustre un convertisseur numérique analogique à 4 entrées

La tension de référence V_{ref} indique la tension maximale de sortie également appelée full-scale output

La résolution dans ce cas est de 1V. Elle correspond au à la variation minimale apparaissant au niveau de la sortie pour un changement des entrées.



D	C	B	A	V_{OUT}	
0	0	0	0	0	Volts
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	
1	0	0	1	9	
1	0	1	0	10	
1	0	1	1	11	
1	1	0	0	12	
1	1	0	1	13	
1	1	1	0	14	
1	1	1	1	15	Volts

(b)

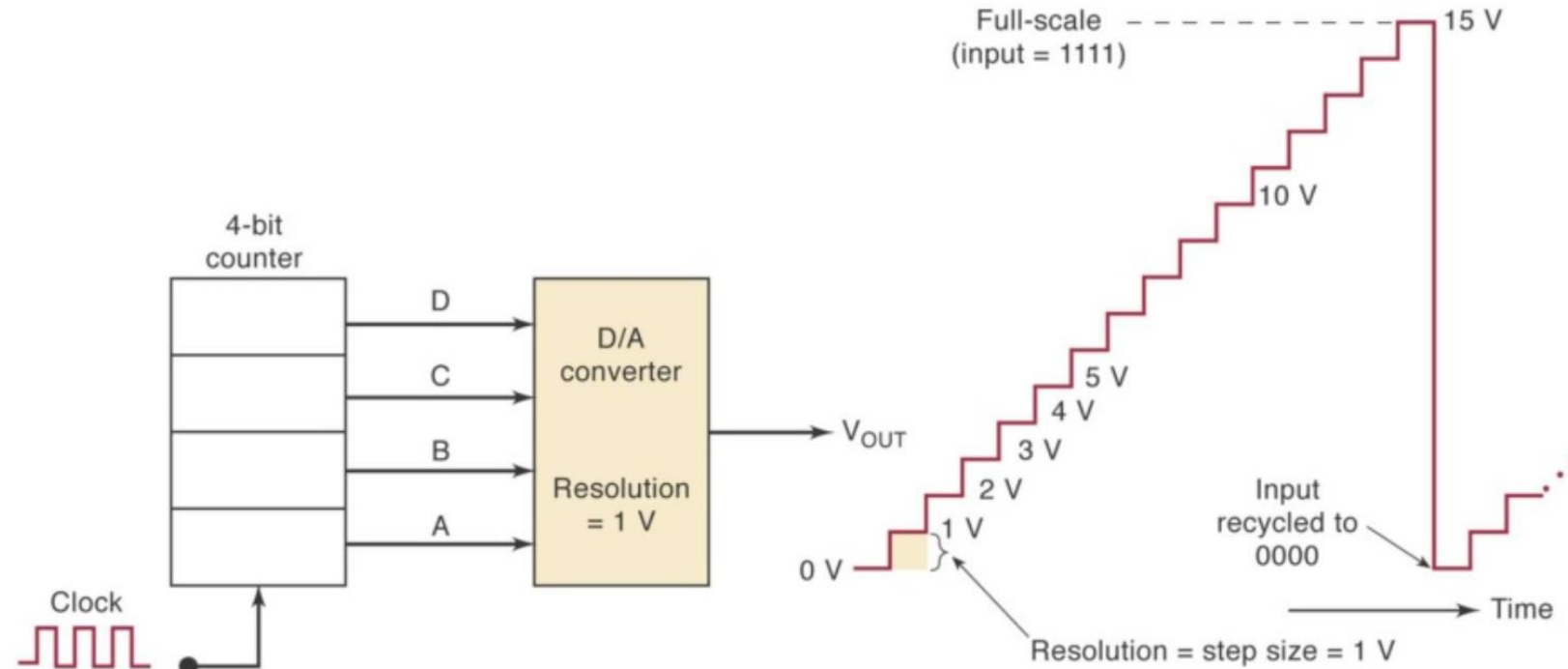
➤ Rôle d'un DAC

- On définit:

- **$N_Level = 2^N$** (N: nombre d'entrées)
- **$N_Step = 2^N - 1$** (N: nombre d'entrées)
- **$Resolution = V_{ref}/N_Step$**

Dans l'exemple ci-joint nous avons:

- **16 niveaux de tensions**
- **15 pas**
- **La résolution = $15/(2^4-1) = 1V$**

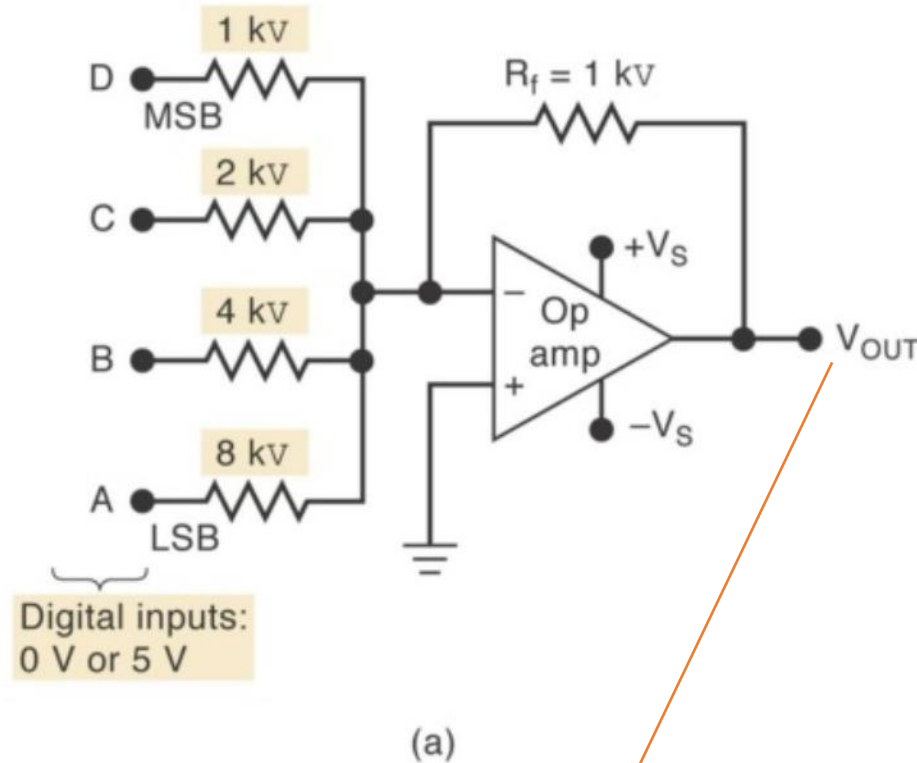


➤ Circuit équivalent simple d'un DAC

Dans l'exemple ci-joint, la résolution est équivalente à

$$9.375/15 = 0.625V$$

Cette résolution est fixée par la valeur de la résistance connectée à la broche A LSB

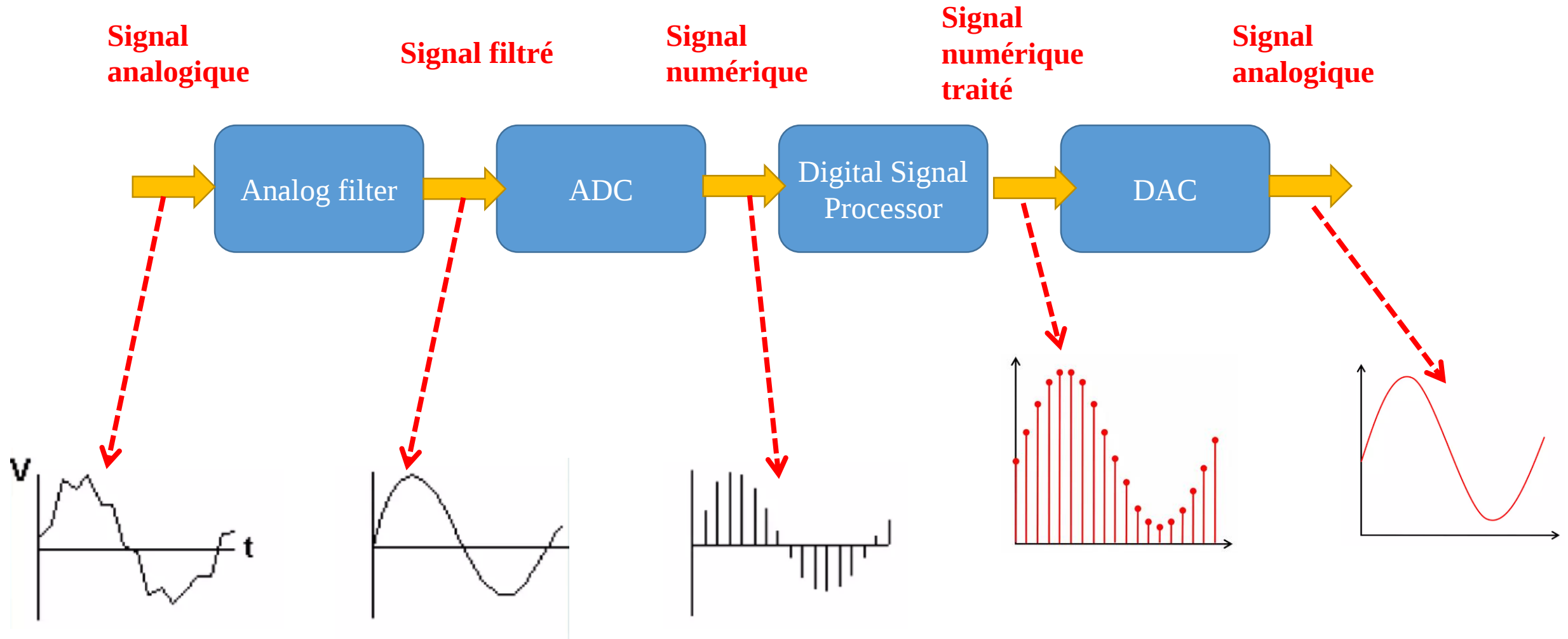


$$V_{out} = - (V_D + \frac{1}{2} V_C + \frac{1}{4} V_B + \frac{1}{8} V_A)$$

Input code				V_{OUT} (volts)
D	C	B	A	
0	0	0	0	0
0	0	0	1	-0.625 ← LSB
0	0	1	0	-1.250
0	0	1	1	-1.875
0	1	0	0	-2.500
0	1	0	1	-3.125
0	1	1	0	-3.750
0	1	1	1	-4.375
1	0	0	0	-5.000
1	0	0	1	-5.625
1	0	1	0	-6.250
1	0	1	1	-6.875
1	1	0	0	-7.500
1	1	0	1	-8.125
1	1	1	0	-8.750
1	1	1	1	-9.375 ← Full-scale

(b)

➤ Chaîne de traitement d'un DSP



➤ Applications des DSP

- ❑ **Industrie** : contrôle commande des processus industriels
- ❑ **Automobile**: contrôle des systèmes automobiles (ACC, ABS, ESP,)
- ❑ **Energie** : contrôle des systèmes de conversion d'énergie (convertisseurs de puissance)
- ❑ **Télécommunication**: modulation/démodulation des signaux
- ❑ **Électronique à usage général**: portable, caméra...
- ❑ **Divers**: scanners, radars, militaire, oscilloscopes, analyseurs de réseau....

➤ Texas Instrument DSP C2000 F28335



- **GPIO** : 88
- **DSP core voltage** 1.8-1.9V with 0- 3.3V input/output
- **DAC** : 4 parallel DAC (8 bits and 12 bits)

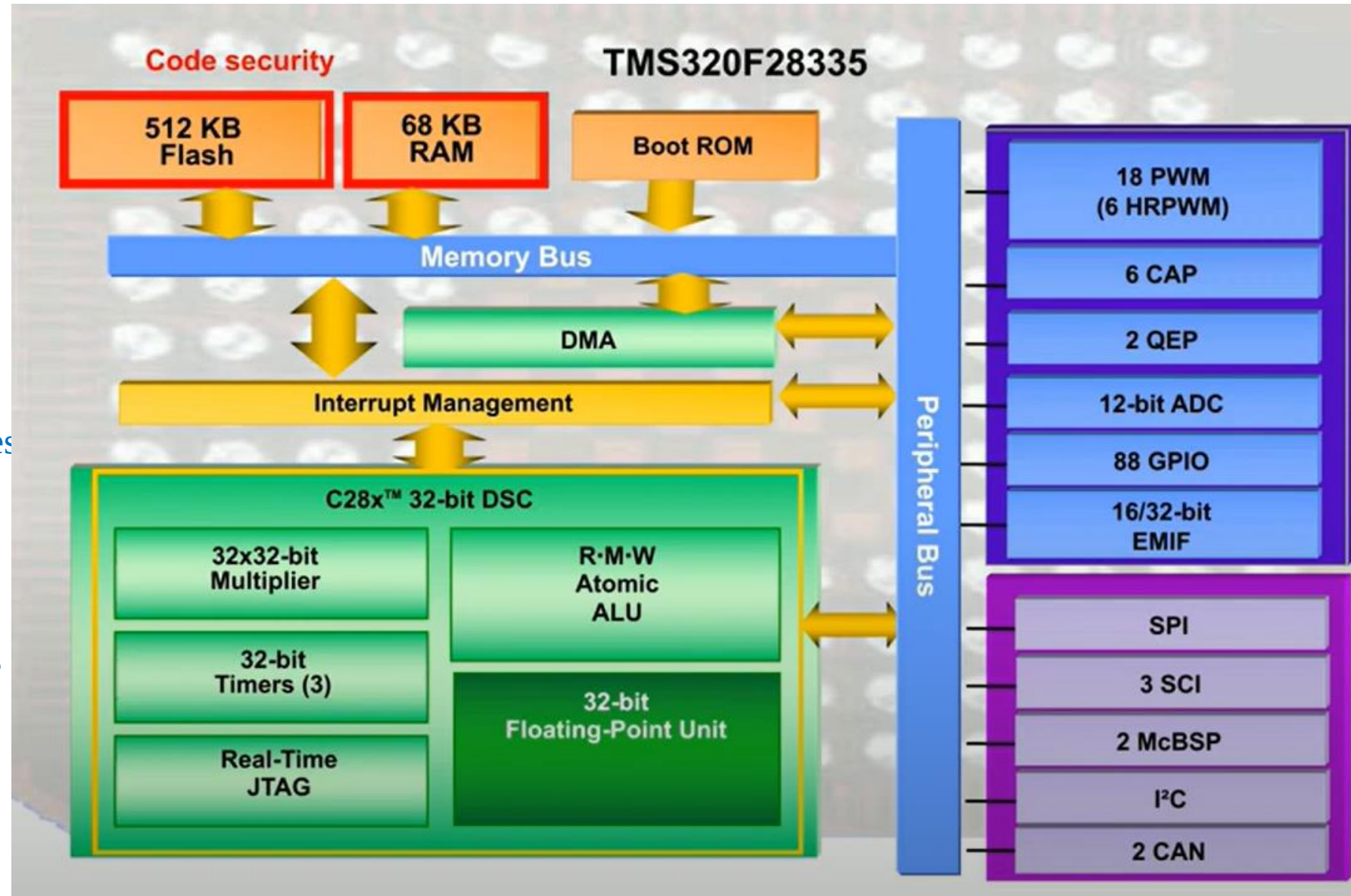
Parameters	Package Pins Size	Features	Description
CPU			C28x
Frequency (MHz)			150
Flash memory (kByte)			512
RAM (kByte)			68
ADC resolution (Bps)			12
Total processing (MIPS)			150
Features			External memory interface, FPU32
UART			3
CAN (#)			2
Sigma-delta filter			0
PWM (Ch)			12
Number of ADC channels			16
Direct memory access (Ch)			6

➤ Architecture globale du DSP C2000 F28335

1- Bloc mémoire:

- **Une mémoire flash** pour le stockage du programme à exécuter
- **Une mémoire RAM** pour le stockage temporaire des données
- **Une mémoire ROM** code qui est exécuté lorsque le DSP est allumé ou réinitialisé.

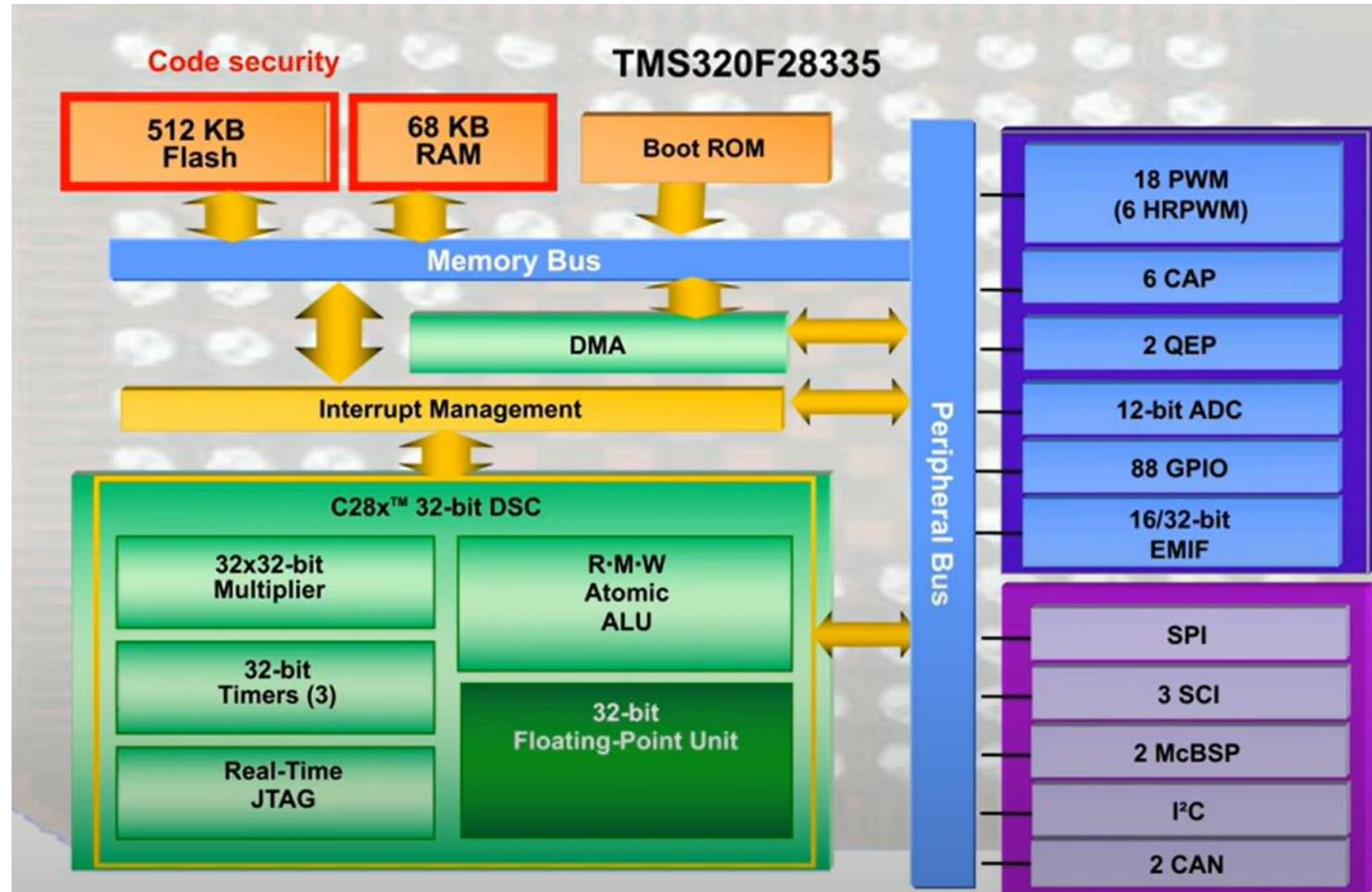
Ce code est responsable de l'initialisation des composants matériels essentiels du DSP, tels que la configuration des registres, la configuration des interfaces mémoire et l'initialisation des périphériques.



➤ Architecture globale du DSP C2000 F28335

2- des bus :

- **Un bus mémoire:** memory bus permettant de communiquer les données entre les différentes mémoires, les périphériques et le processeur central
- **Un bus de périphériques** permettant la liaison entre les différentes périphériques d'entrées/sorties, les mémoires et l'unité de centrale de traitement



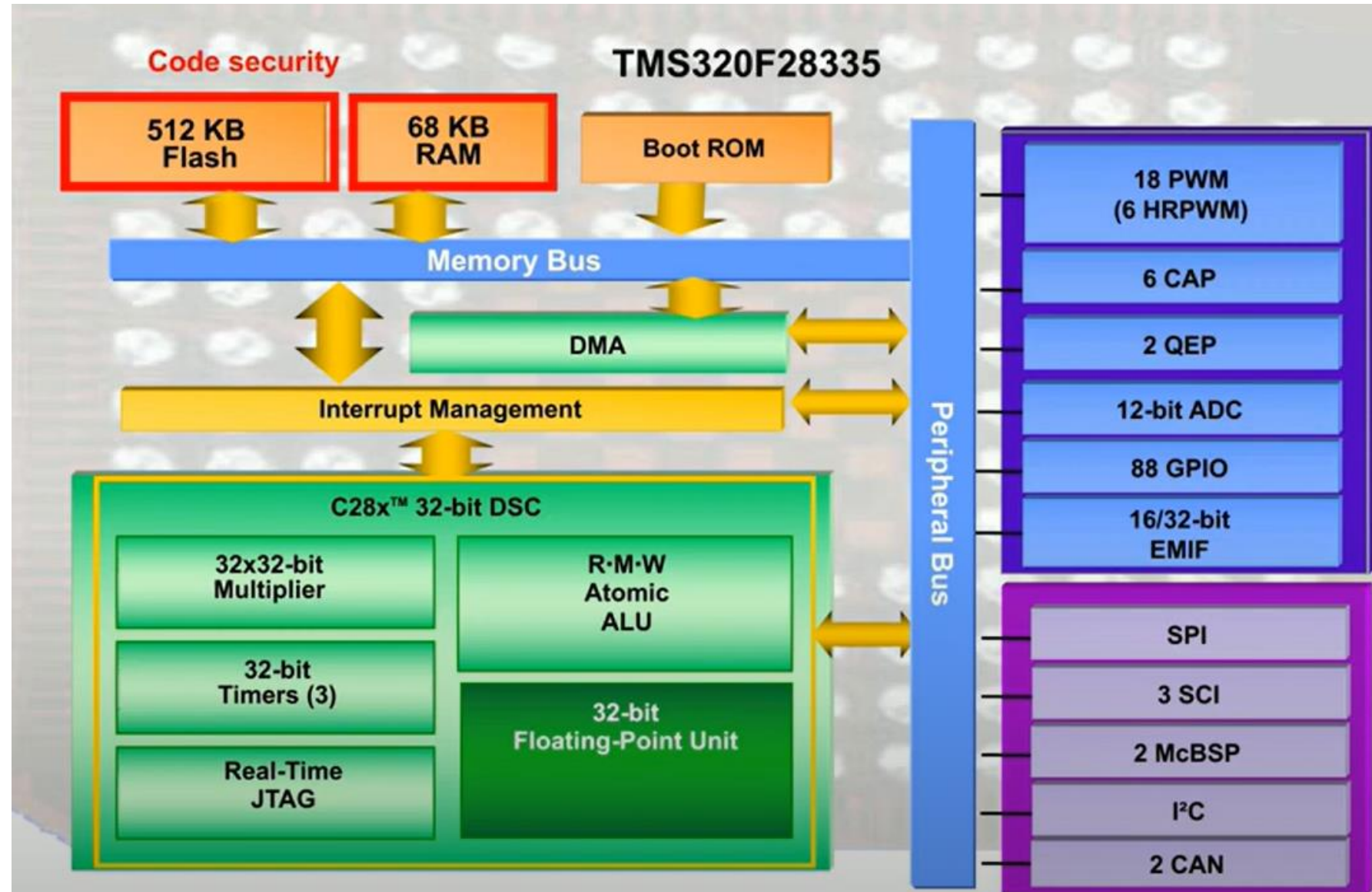
➤ Architecture globale du DSP C2000 F28335

3- Un DMA Controller:

DMA: Accès direct à la mémoire (DMA) :

Le DMA est une fonctionnalité qui permet aux périphériques de communiquer directement avec la mémoire du système sans impliquer l'unité centrale de traitement (CPU).

Cela contribue à améliorer l'efficacité des transferts de données.

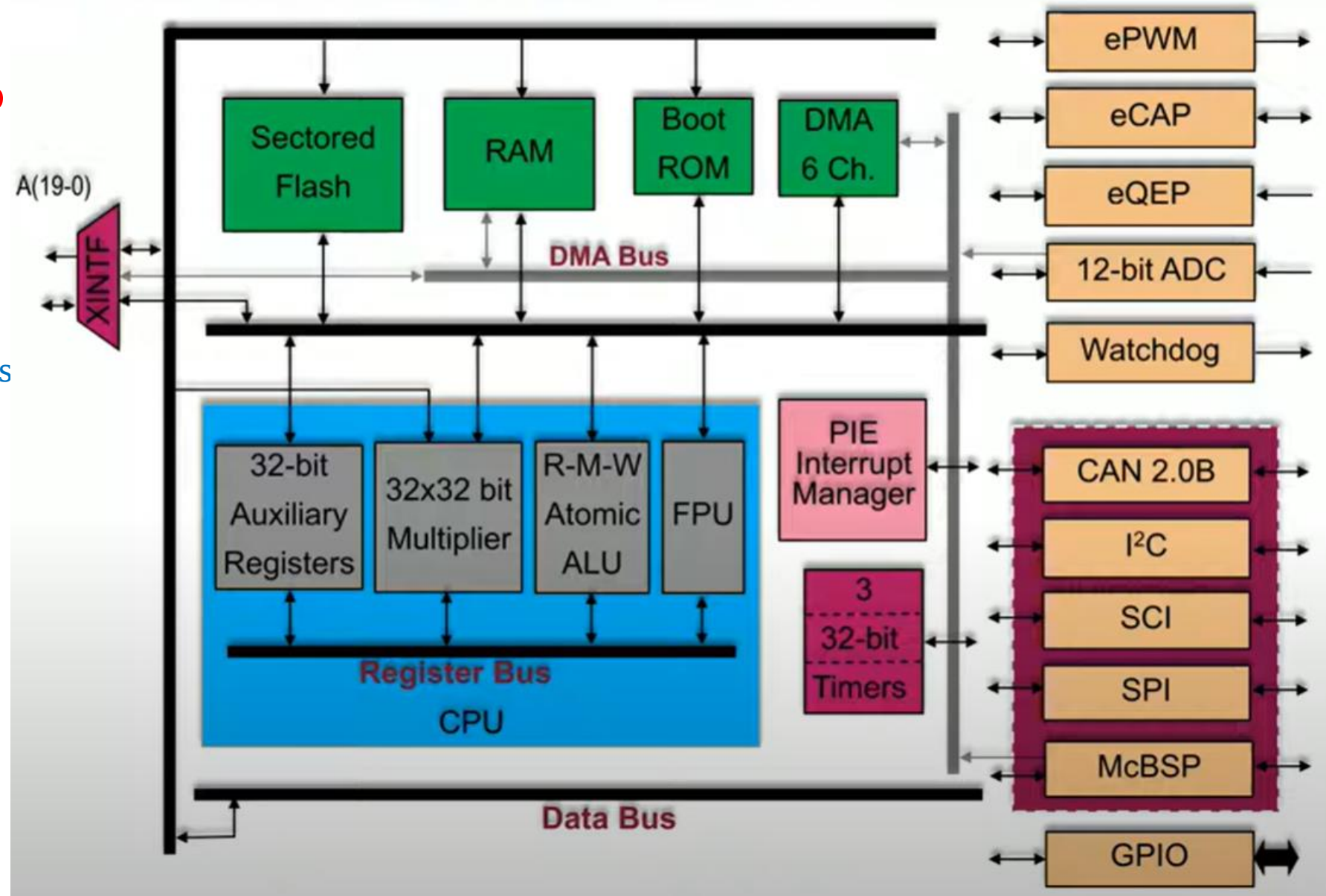


➤ Architecture globale du DSP C2000 F28335

4- CPU:

Registres auxiliaires: (Status Register) :Contient des indicateurs de l'état actuel du CPU, tels que les indicateurs de drapeau (flag) pour les opérations arithmétiques (positif, négatif, zéro, retenu, etc.).

Multiplicateur de 32 bits: Son rôle principal est d'effectuer des opérations de multiplication sur des données numériques

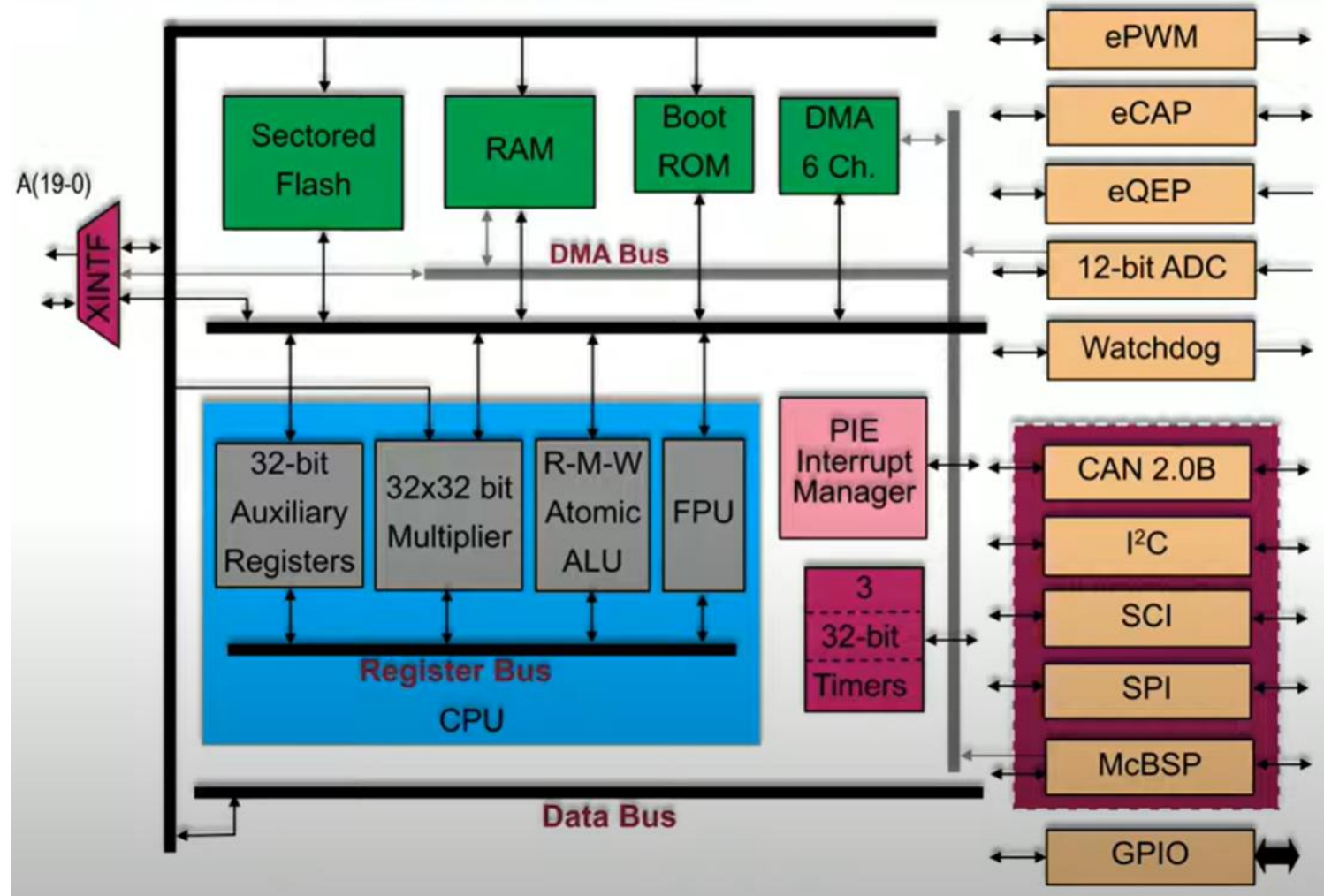


➤ Architecture globale du DSP C2000 F28335

4- CPU:

Atomic ALU: une « opération atomique » fait référence à une opération qui semble être exécutée comme une unité unique et indivisible, sans interférence d'autres opérations. Dans un environnement multitraitement, les opérations atomiques sont souvent utilisées pour garantir la cohérence des données et éviter les conditions de concurrence.

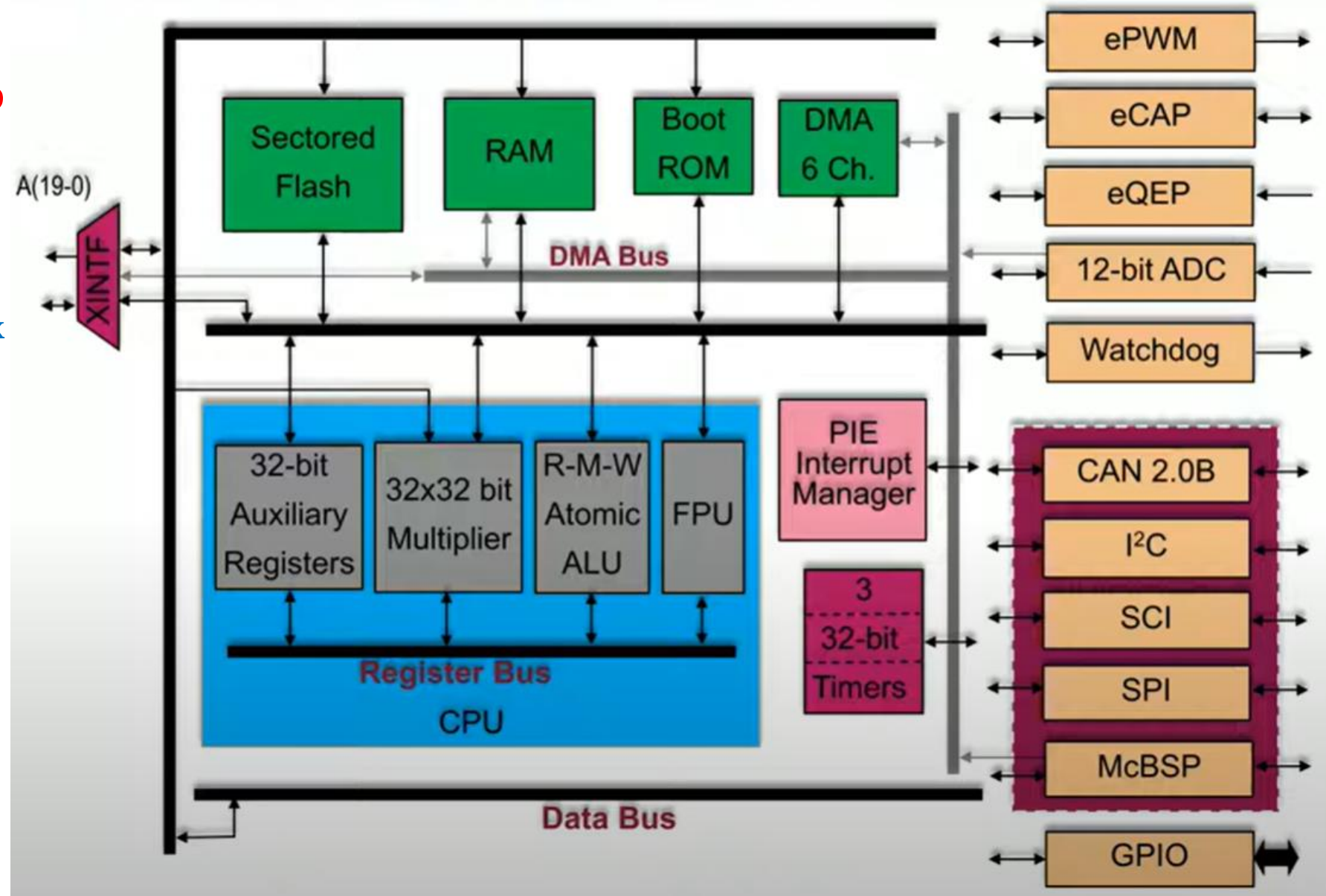
FPU Floating Point Unit: L'objectif principal d'un FPU est d'effectuer des opérations arithmétiques sur des nombres à virgule flottante. Cela inclut des opérations telles que l'addition, la soustraction, la multiplication et la division impliquant des nombres avec des points décimaux.



➤ Architecture globale du DSP C2000 F28335

PIE Interrupt Manager:

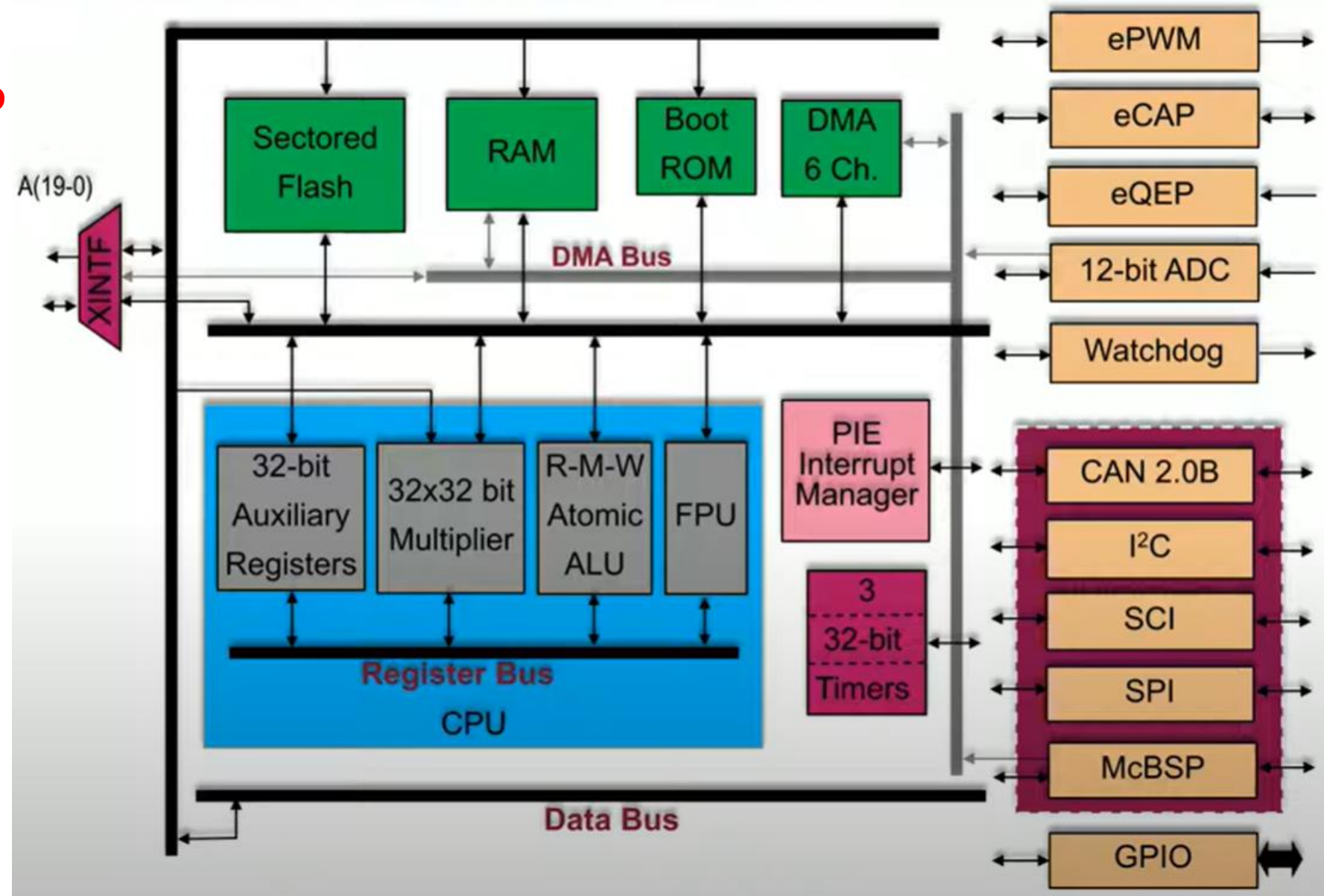
Les interruptions sont des signaux envoyés par le matériel ou le logiciel au processeur CPU pour demander son attention. Un gestionnaire d'interruptions organise et hiérarchise ces interruptions..



➤ Architecture globale du DSP C2000 F28335

eCAP (Enhanced Capture) :

eCAP est principalement conçu pour capturer l'horodatage ou la durée des événements externes. Il permet au DSP de capturer le temps exact à laquelle un événement spécifique se produit. Il permet également de synchroniser les opérations sur des fronts montants ou descendants d'un signal.

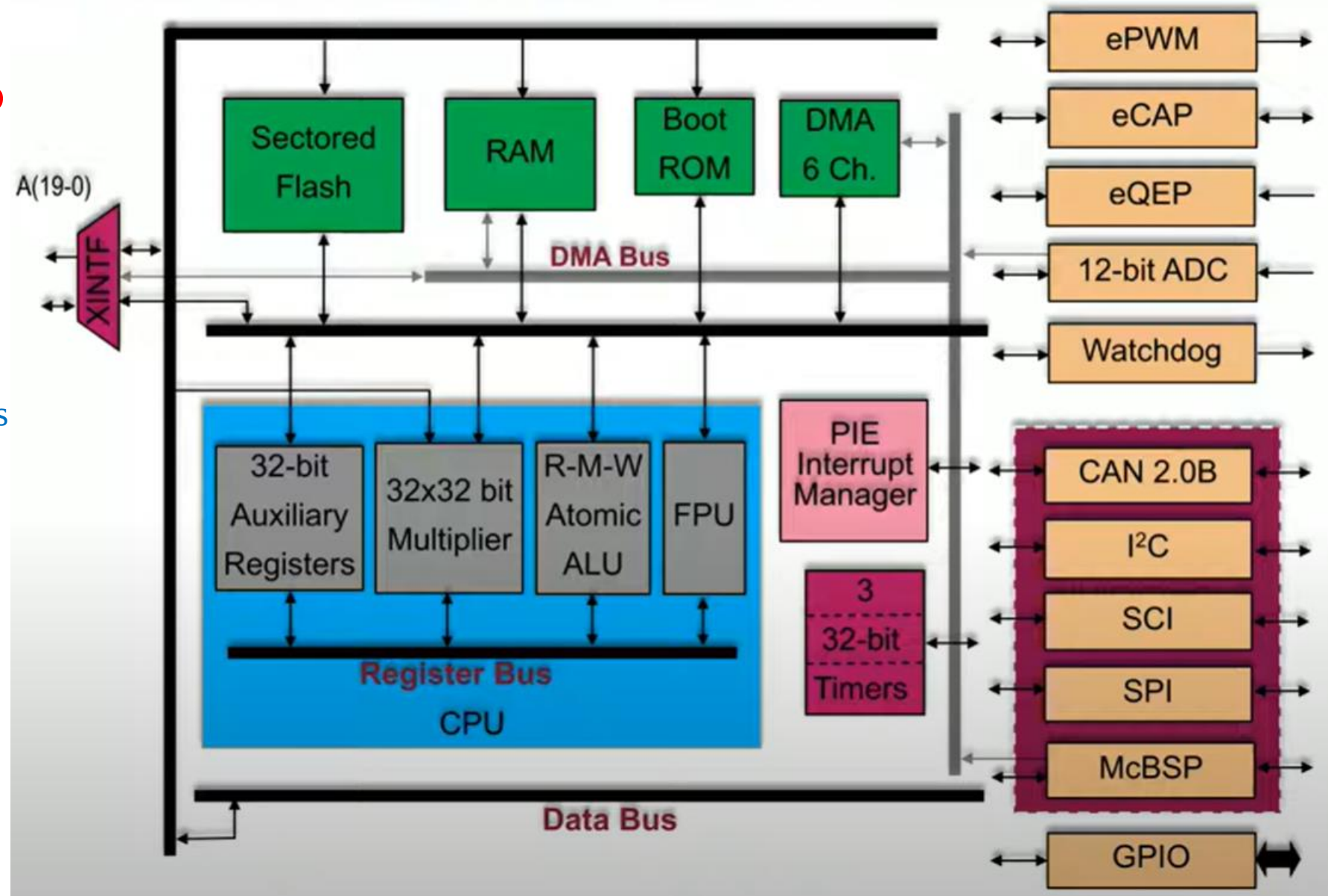


➤ Architecture globale du DSP C2000 F28335

eQEP (Enhanced Quadrature Encoder Processing) :

La fonction principale de l'eQEP est de s'interfacer avec les codeurs en quadrature.

Les codeurs en quadrature génèrent deux signaux (A et B) qui ont une relation de phase spécifique, permettant de déterminer à la fois la position et le sens de rotation d'un moteur



➤ Architecture globale du DSP C2000 F28335

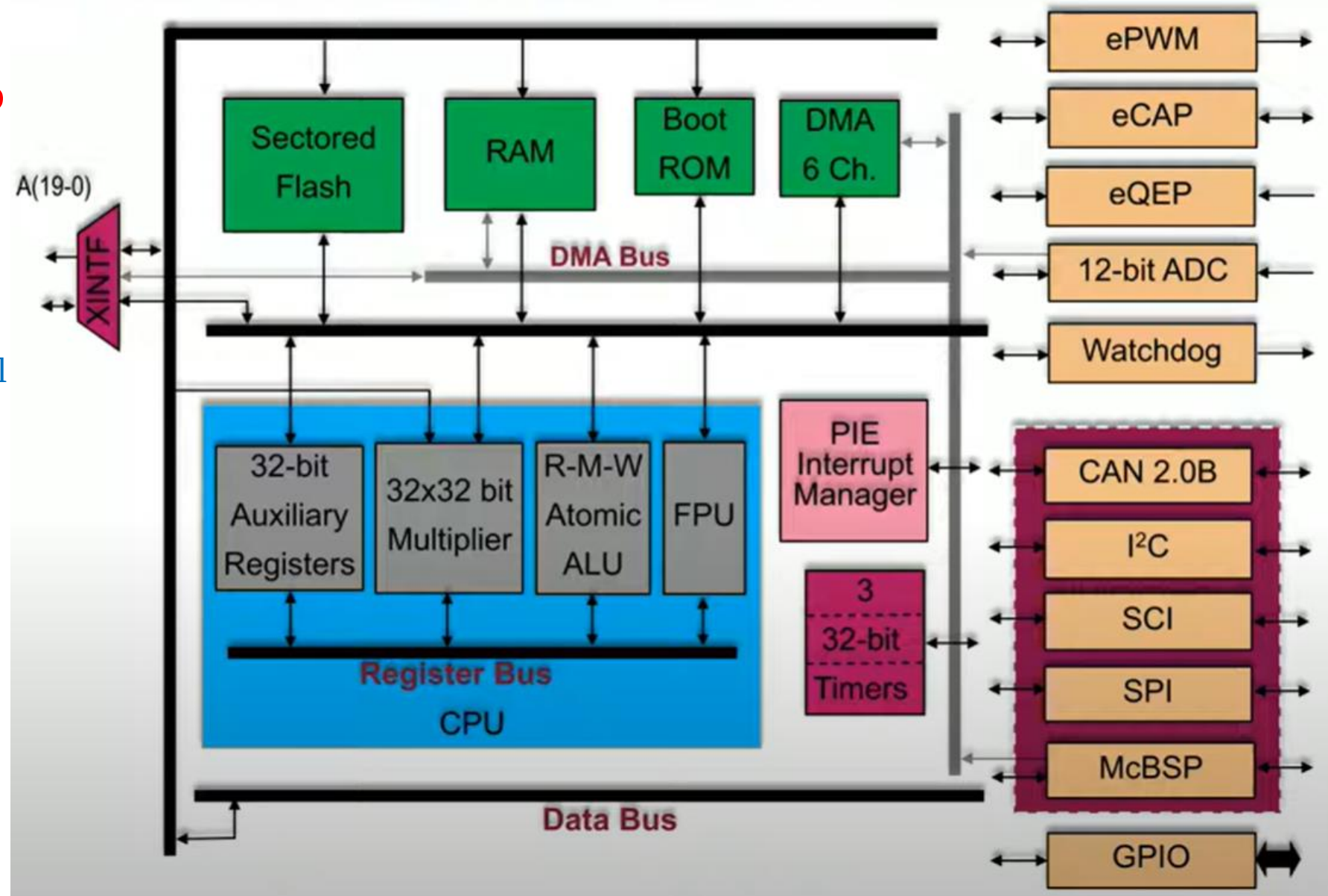
12-Bits ADC (Analog to Digital Converter):

Consiste à échantillonner le signal analogique à des intervalles spécifiques et à attribuer un code numérique à chaque échantillon.

Résolution (nombre de bits dans le signal de sortie): 12 bits (4096 possibilités)

Marge de tension : 0 – 3.3V

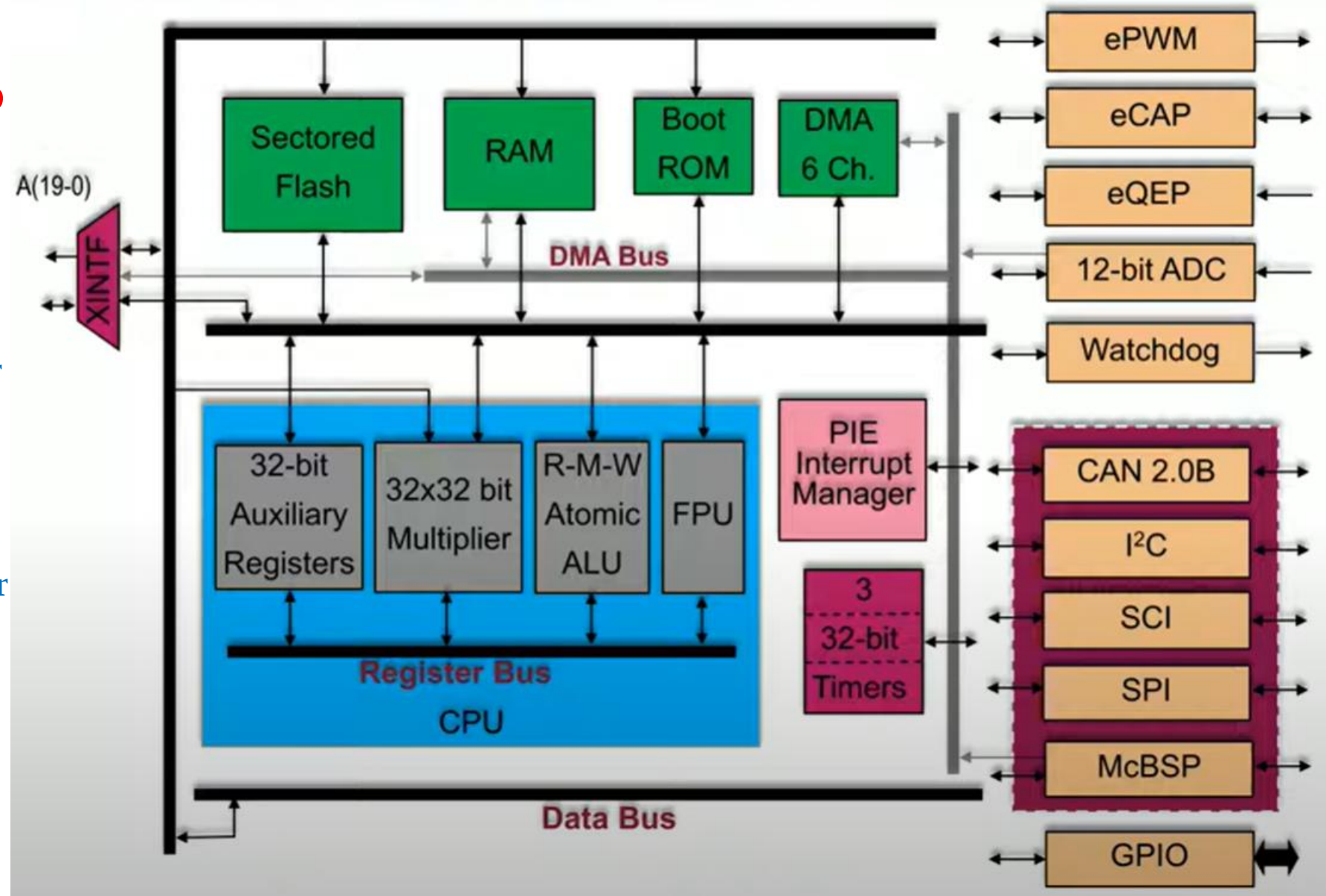
Nombre de chaines :16



➤ Architecture globale du DSP C2000 F28335

Watchdog

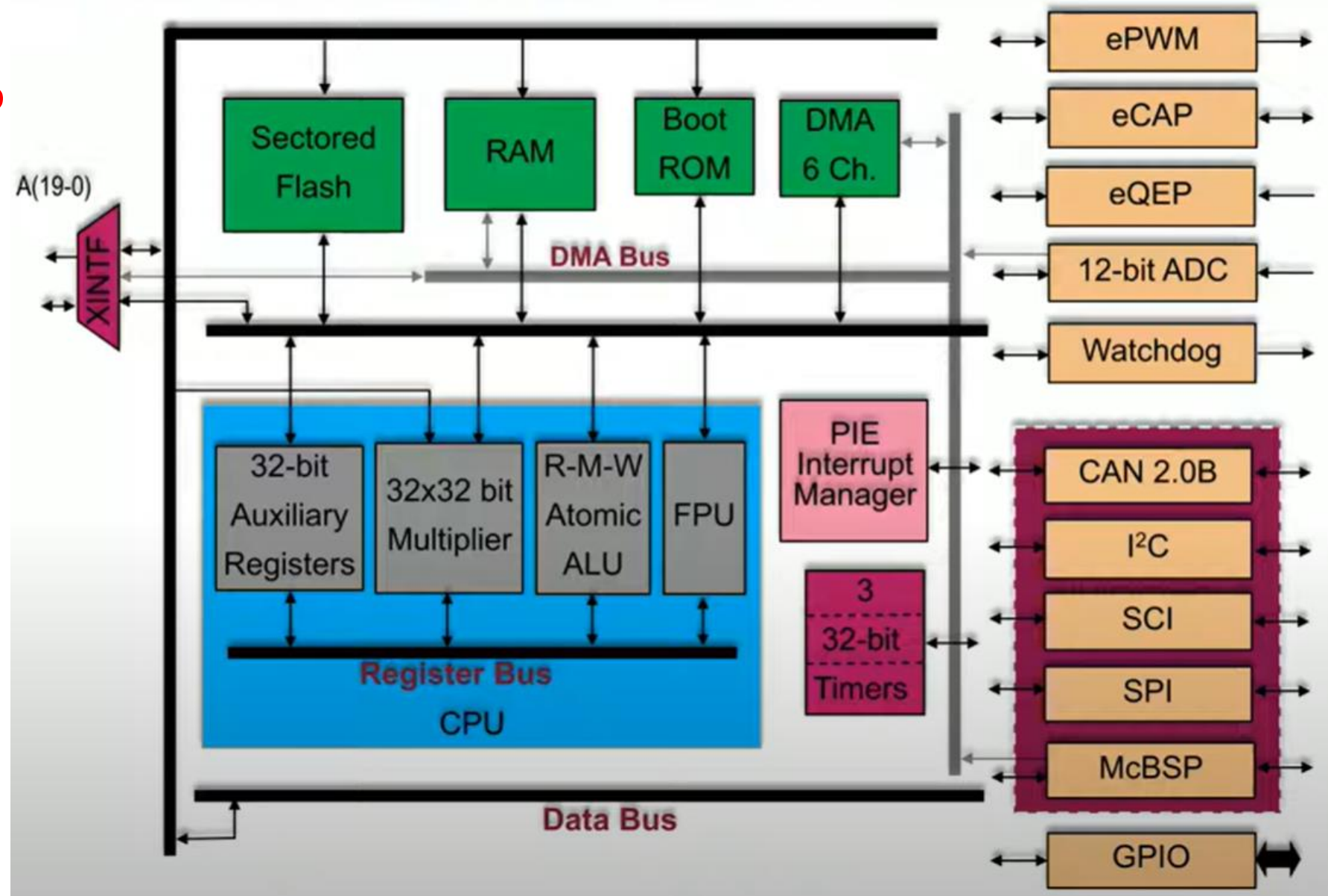
Le rôle principal d'un chien de garde dans un DSP est de détecter les défauts ou anomalies dans le système. Cela inclut l'identification des situations dans lesquelles le DSP n'exécute pas son programme comme prévu, par exemple lorsqu'il reste coincé dans une boucle ou cesse de répondre.



➤ Architecture globale du DSP C2000 F28335

Bus CAN

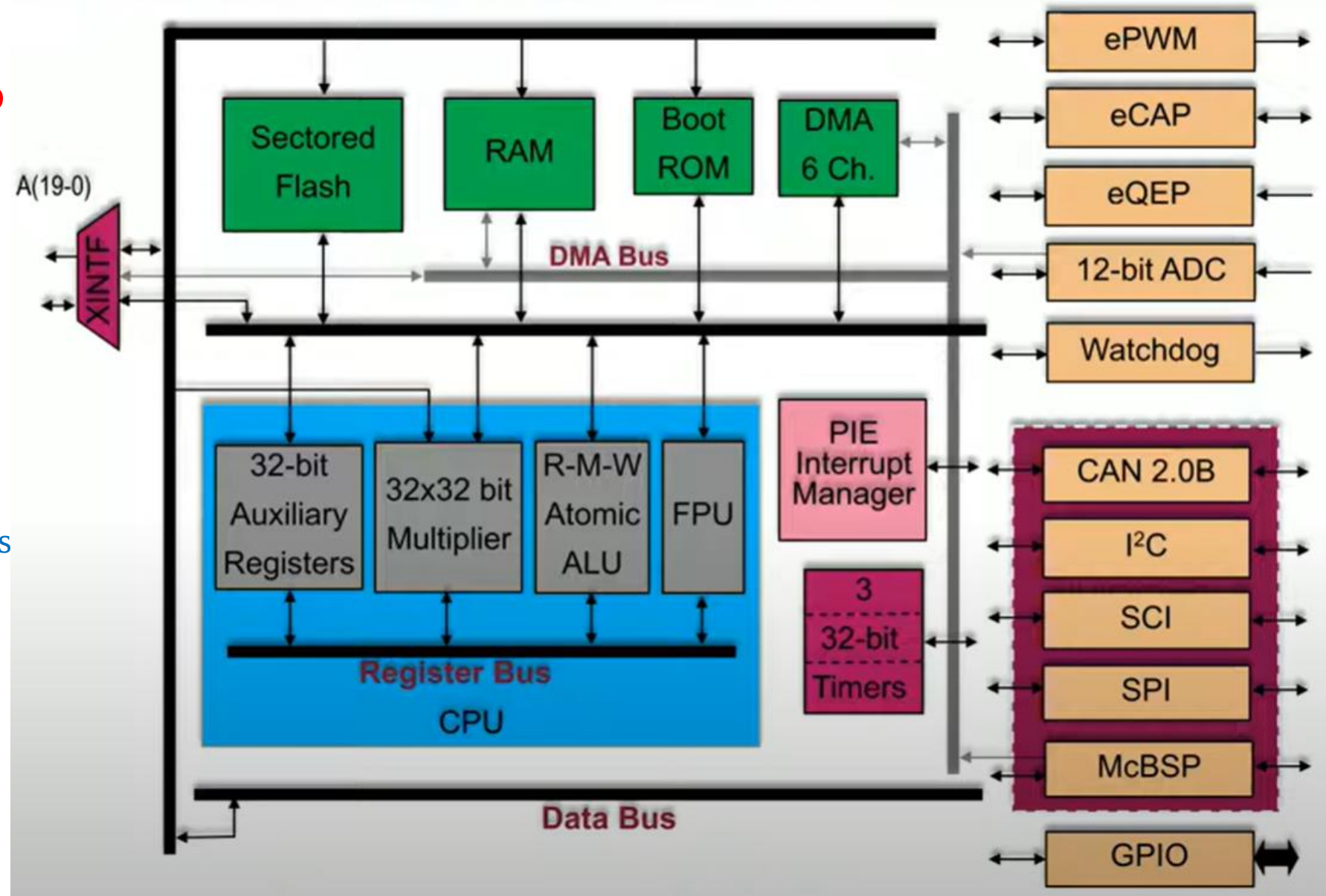
Le terme "CAN" fait référence au Controller Area Network, un protocole de communication largement utilisé dans les systèmes embarqués pour la communication entre différents modules électroniques.



➤ Architecture globale du DSP C2000 F28335

Bus I2C (Inter Integrated Circuit)

L'I2C est un protocole de communication série bidirectionnelle qui permet au DSP d'échanger des données avec d'autres périphériques compatibles I2C, tels que des capteurs, des mémoires, des convertisseurs analogique-numérique (CAN)

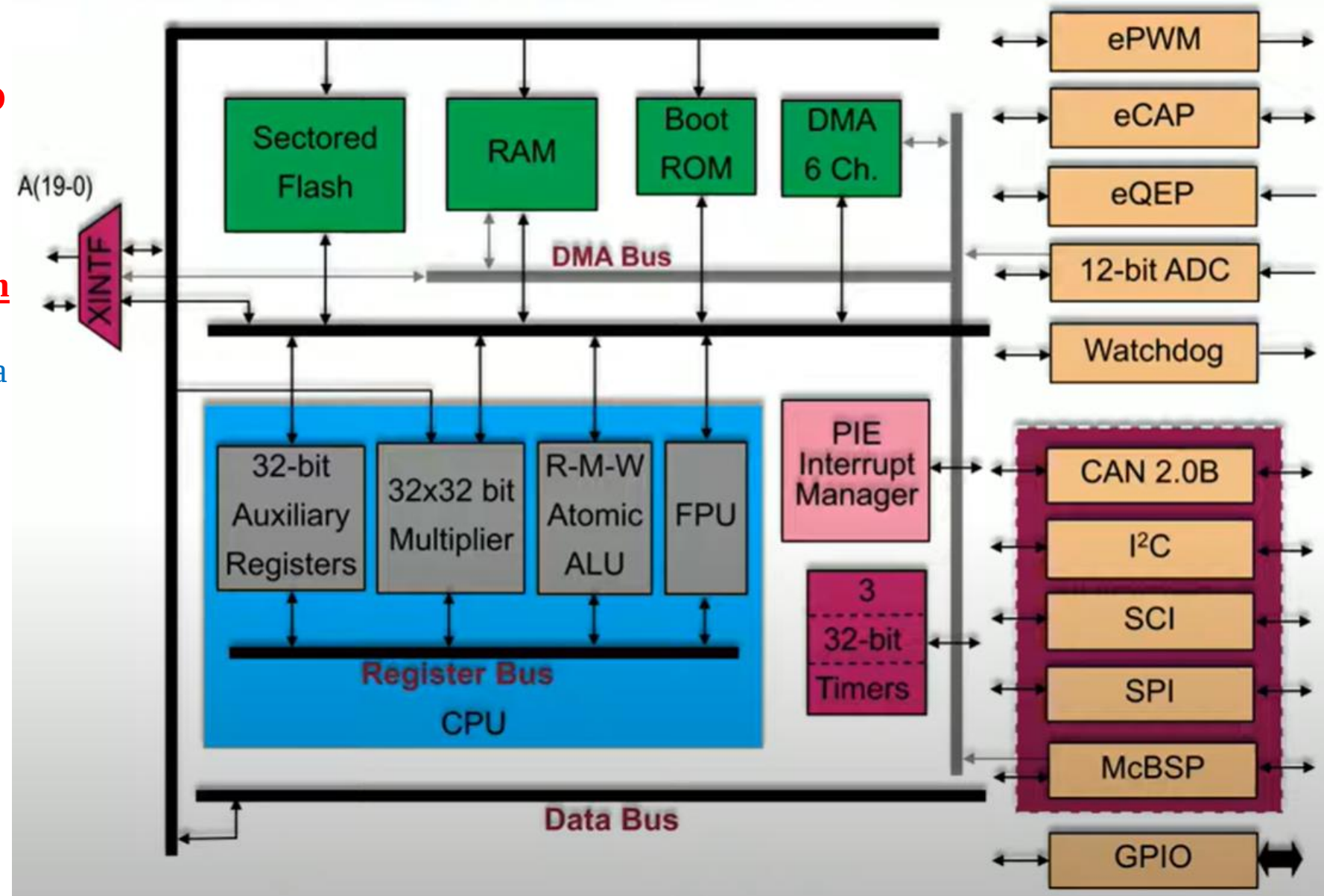


➤ Architecture globale du DSP C2000 F28335

Bus SCI (Serial Communication Interface)

L'interface SCI prend en charge la communication série, qui peut être:

- **asynchrone (UART - Universal Asynchronous Receiver/Transmitter)**
- **ou synchrone (SPI - Serial Peripheral Interface)** en fonction de la configuration.

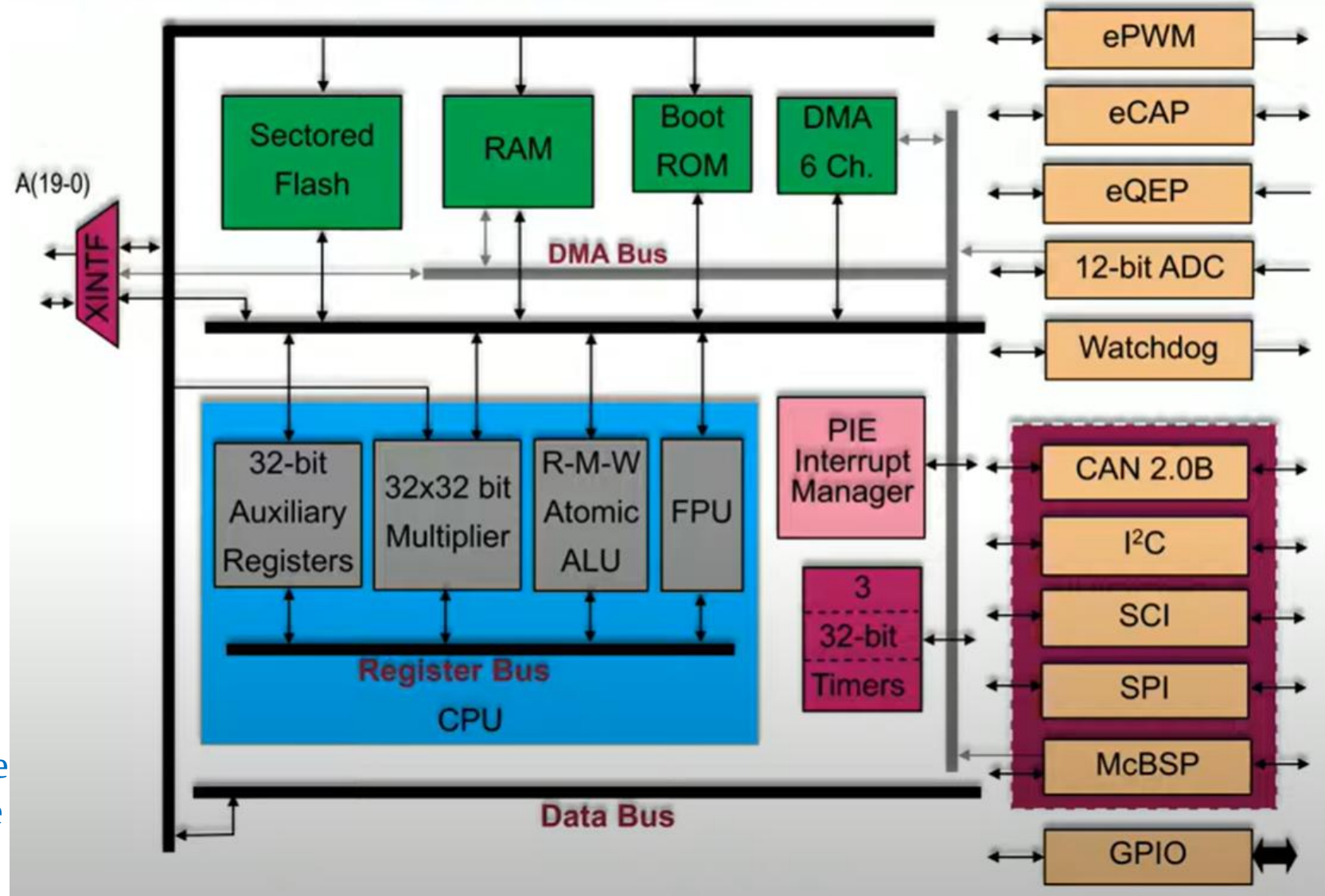


➤ Architecture globale du DSP C2000 F28335

McBSP (Multi-Channel Buffered Serial Ports)

Contrairement à d'autres interfaces série simples, la McBSP est conçue pour prendre en charge plusieurs canaux de communication, ce qui peut être utile dans des applications nécessitant des transferts de données simultanés sur plusieurs canaux.

La McBSP est souvent utilisée dans des **applications audio numériques**, où elle peut être utilisée pour la transmission de données audio entre le DSP et des convertisseurs analogique-numérique (ADC) ou numérique-analogique (DAC)

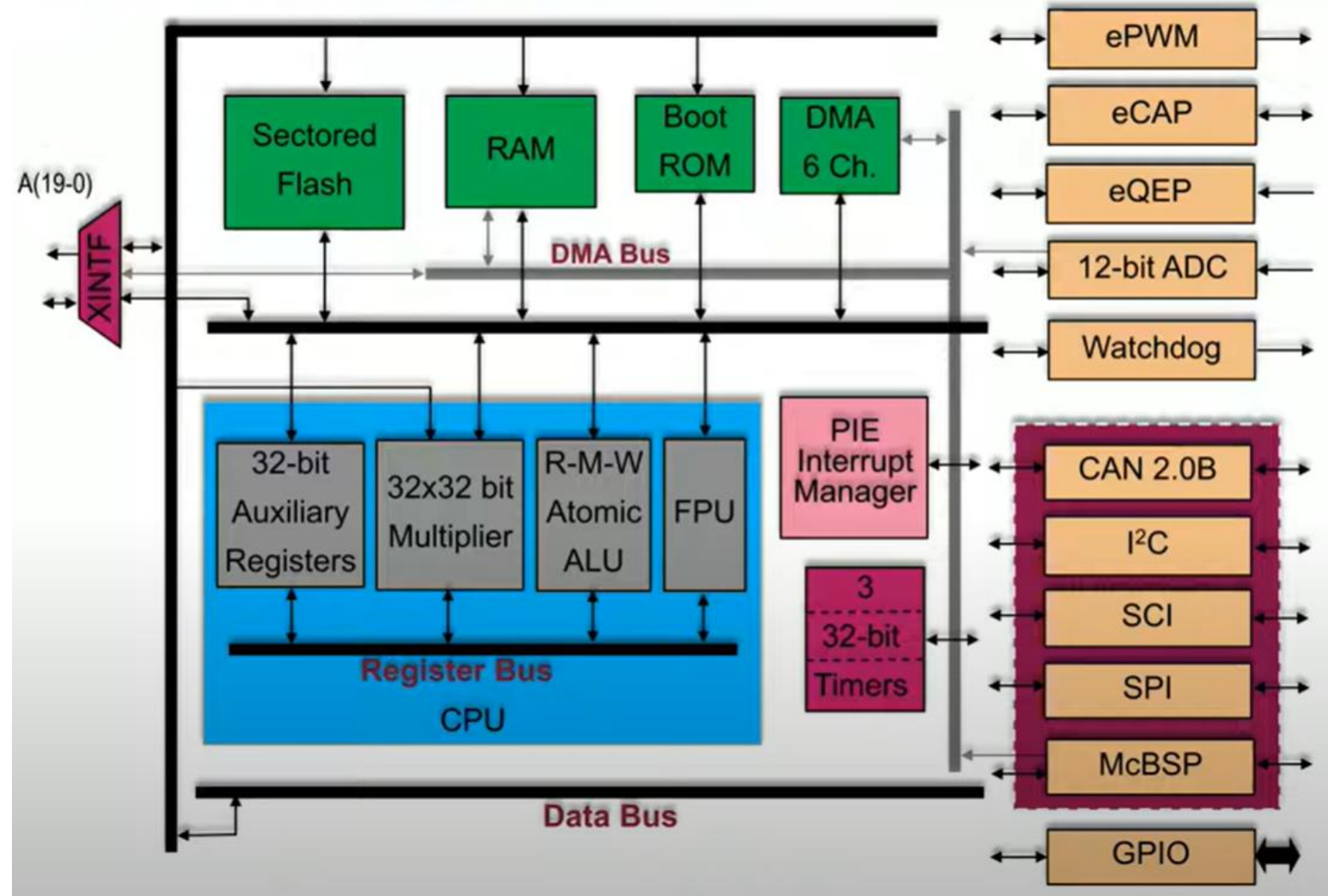


➤ Architecture globale du DSP C2000 F28335

GPIO : Genral Purpose Input and Output

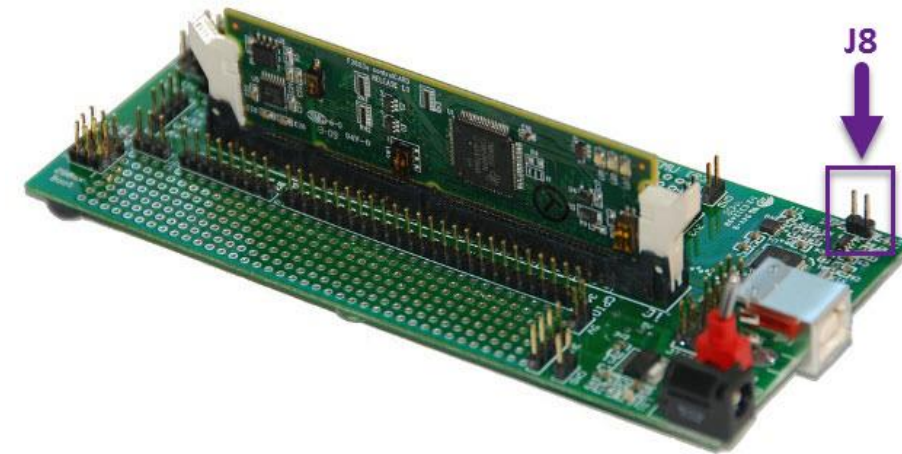
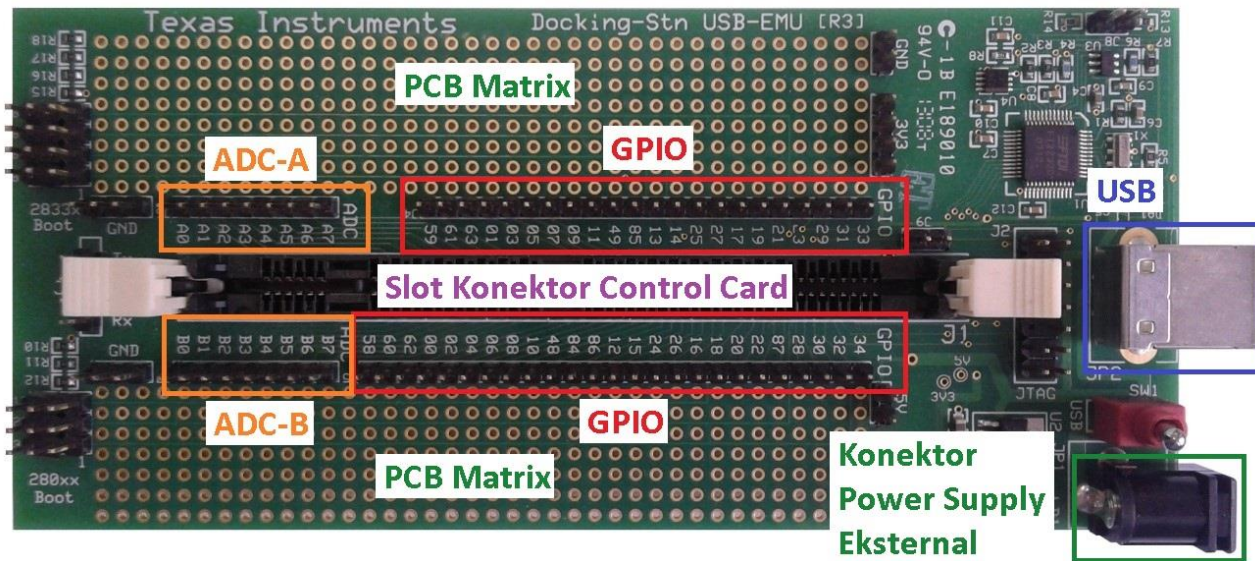
Les GPIO (General-Purpose Input/Output) sont des broches ou des ports configurables sur un DSP ou tout autre dispositif embarqué, permettant à ce dernier de communiquer avec le monde extérieur.

Les GPIO peuvent être configurées en tant qu'entrées ou sorties numériques pour lire ou générer des signaux



➤ Programmation et manipulation de la C2000 F28335

- La carte d'expérimentation (docking station) permet une manipulation facile de la carte C2000F28335.
- Equipée d'un slot permettant l'insertion facile de la carte
- La docking station donne l'accès aux différents ports de la carte C2000 à savoir les ports ADC, GPIO, les interfaces de communications...
- L'interface USB est utilisée soit pour alimenter la carte, soit pour le transfert du code

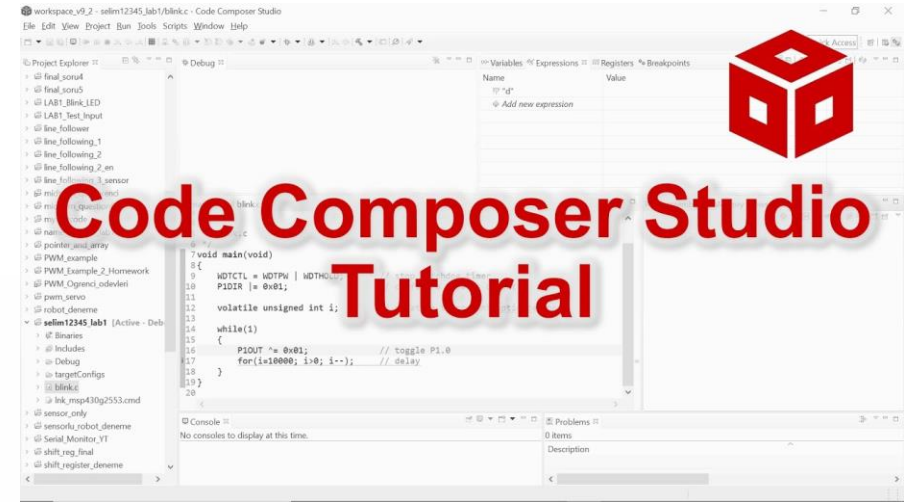


➤ Programmation et manipulation de la C2000 F28335

- La programmation peut se faire soit par le logiciels Code Composer Studio soit par Matlab/Simulink



<https://www.mathworks.com/help/install/ug/download-without-installing.html>

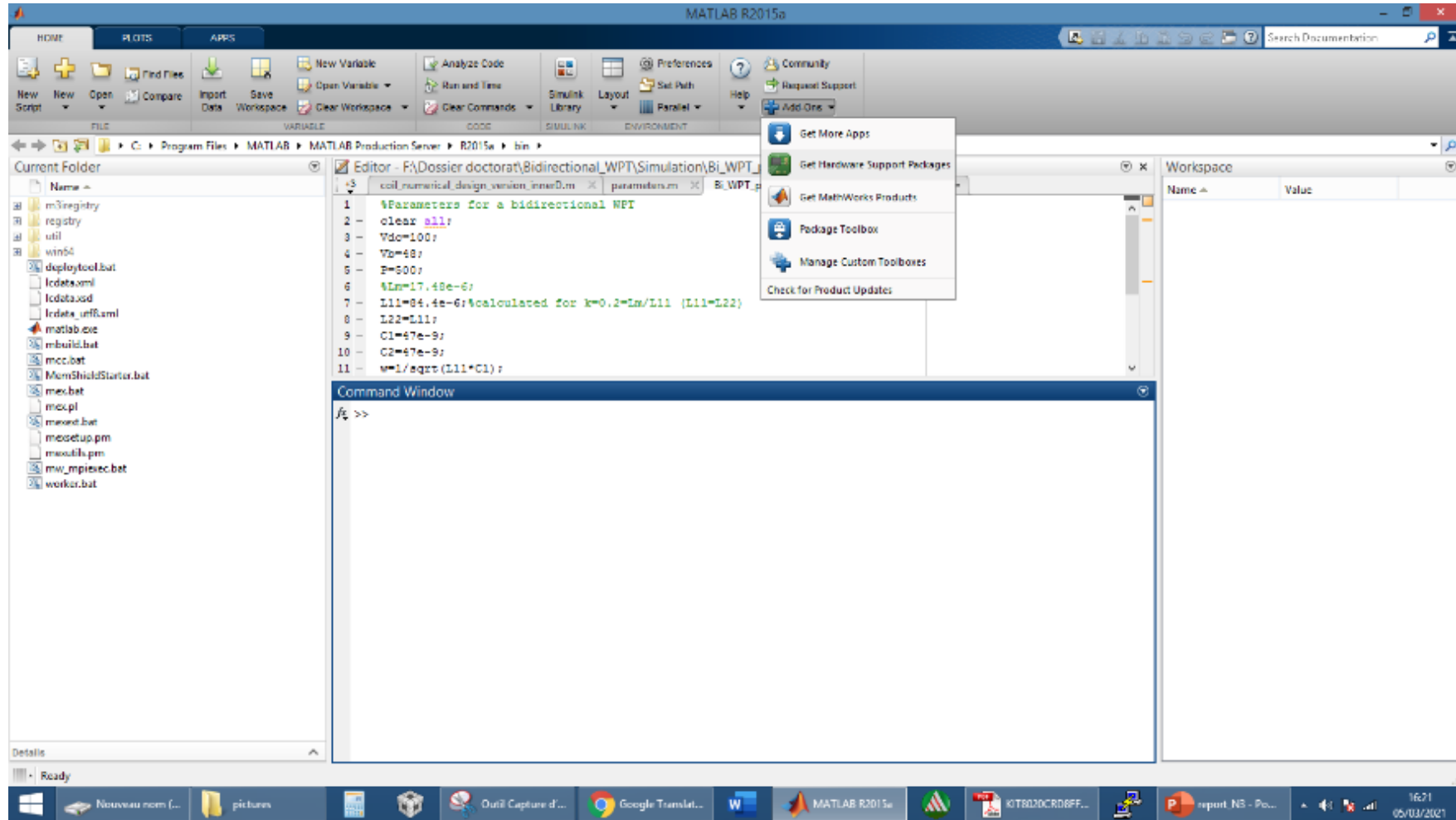


<https://www.ti.com/tool/CCSTUDIO>

➤ Programmation de la C2000 F28335 avec MATLAB

■ Installation des packages et configuration

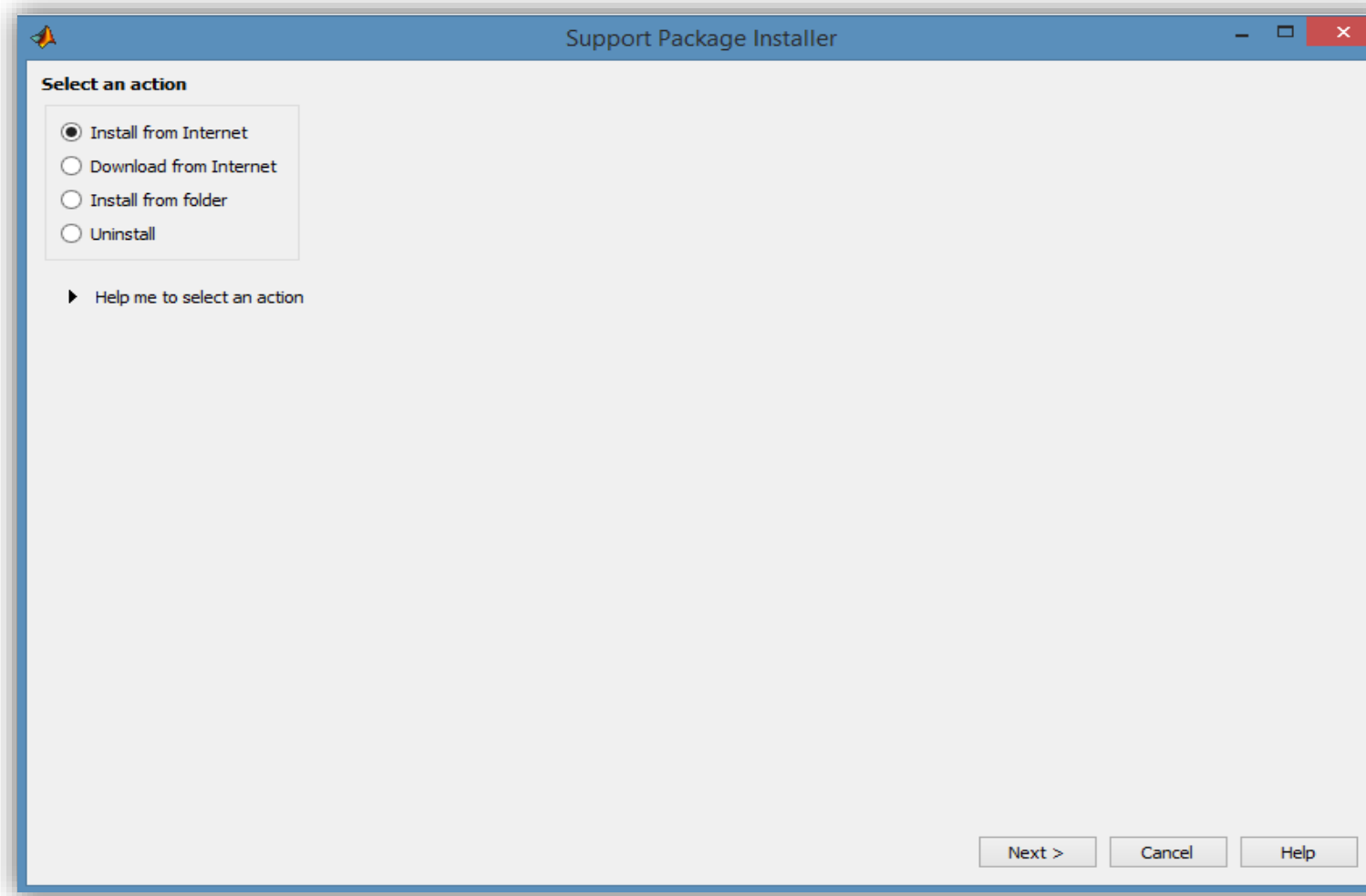
1- In the MATLAB home window, go to **Adds-ons** and choose **Get hardware Supports Packages**. See figure bellow.



➤ Programmation de la C2000 F28335 avec MATLAB

- Installation des packages et configuration

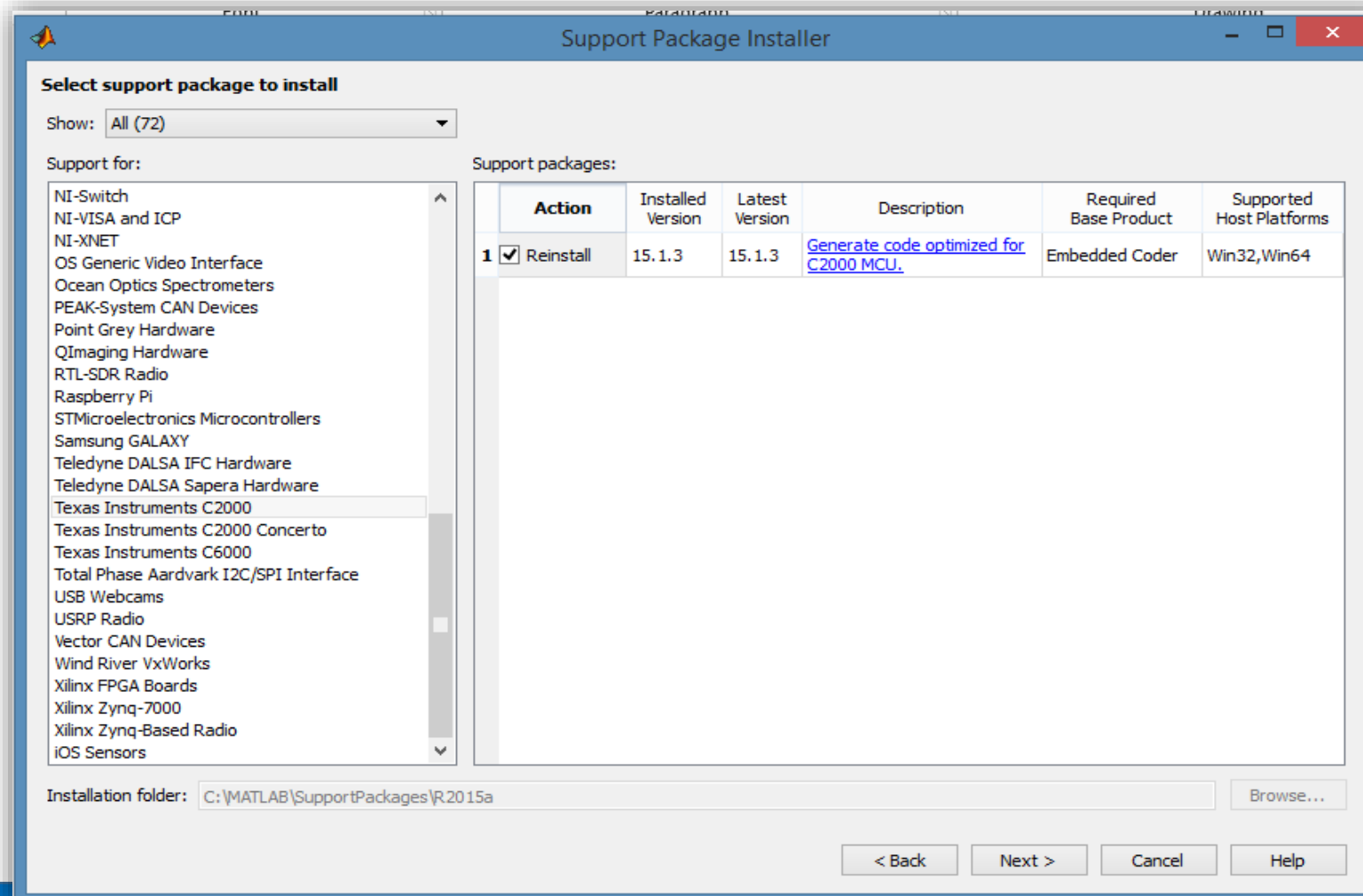
2 - Dans la fenêtre suivante, choisir **Install from Internet**



➤ Programmation de la C2000 F28335 avec MATLAB

■ Installation des packages et configuration

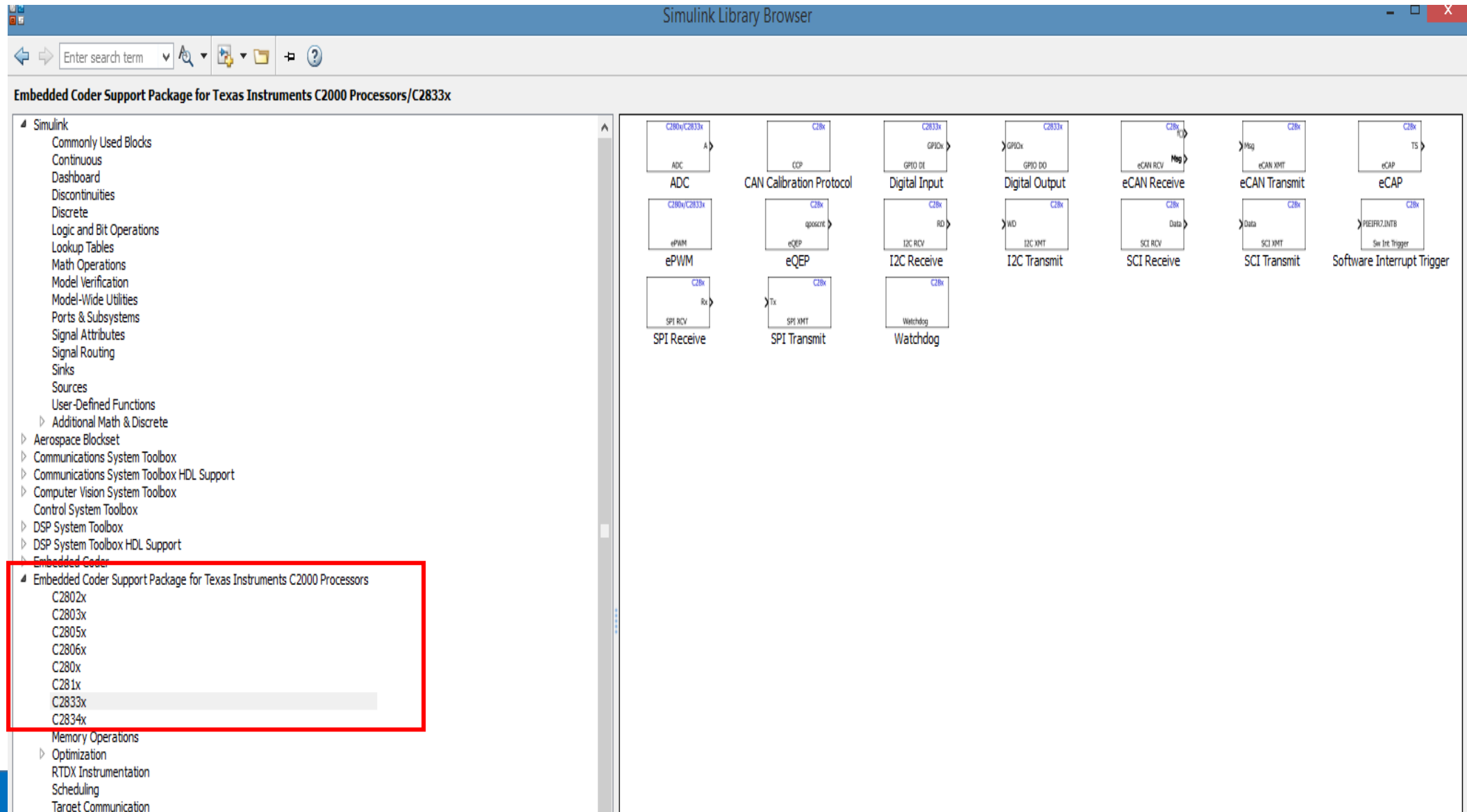
3 - Choisir **Texas Instruments C2000** dans la liste. Puis cliquer sur **Next**.



➤ Programmation de la C2000 F28335 avec MATLAB

■ Installation des packages et configuration

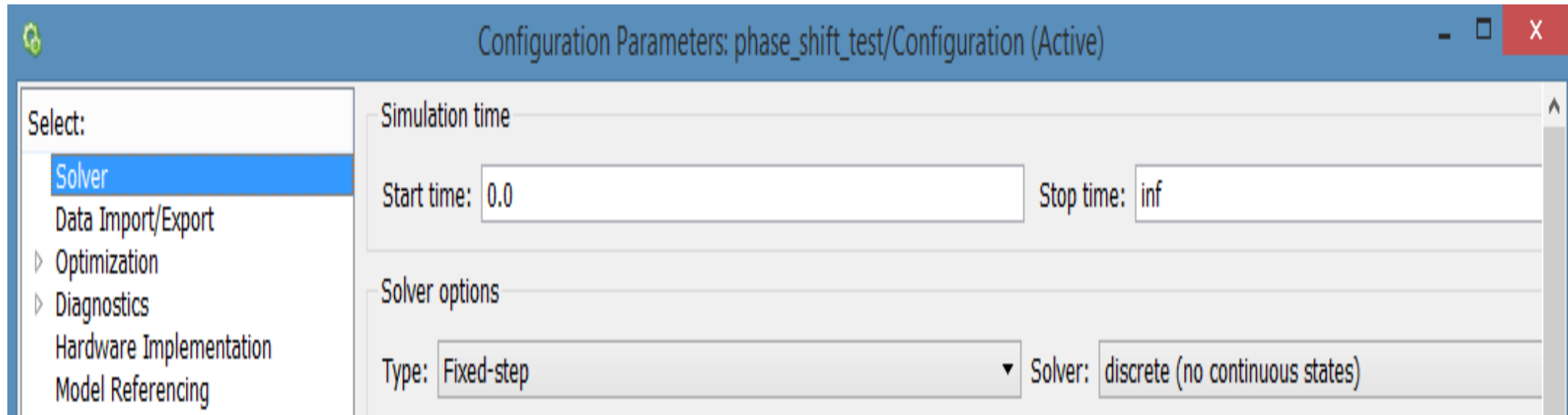
4 - Une fois l'installation terminée, lancer MATLAB / SIMULINK puis vérifier l'apparition des packages installés dans la bibliothèque



➤ Programmation de la C2000 F28335 avec MATLAB

- Installation des packages et configuration

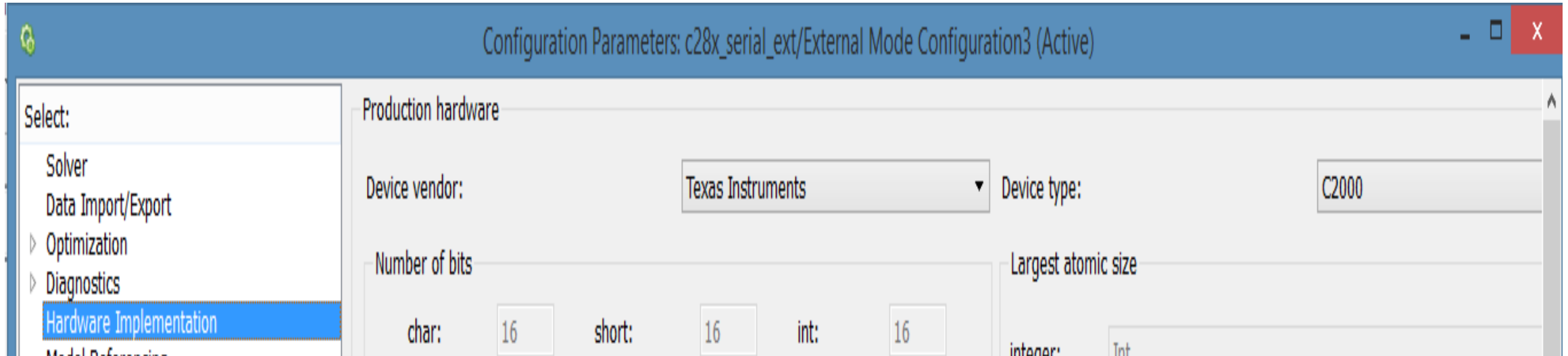
5- Passons maintenant à la configuration Simulink, accédez aux **paramètres de configuration** du modèle, sous les **options du solveur**, choisissez **Discrete Fixed Step** car le C2000 F28335 ne prend pas en charge les solveurs continus.



➤ Programmation de la C2000 F28335 avec MATLAB

- Installation des packages et configuration

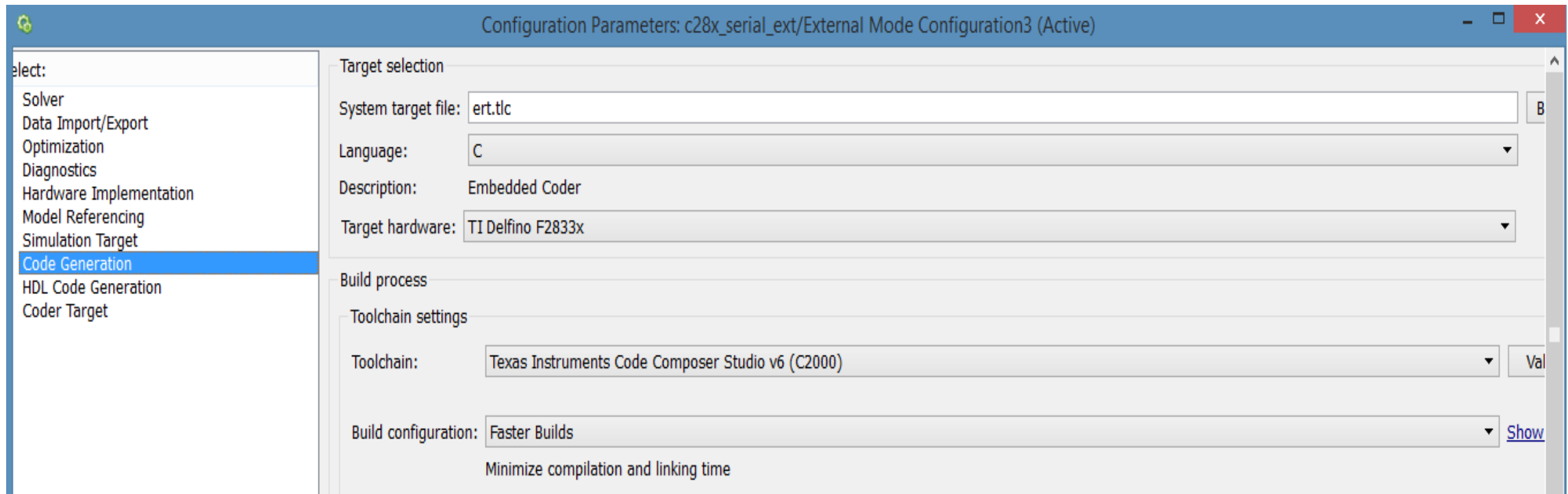
6- aller vers **Hardware Implementation** et choisir **Texas Instrument** puis la carte **C2000**



➤ Programmation de la C2000 F28335 avec MATLAB

▪ Installation des packages et configuration

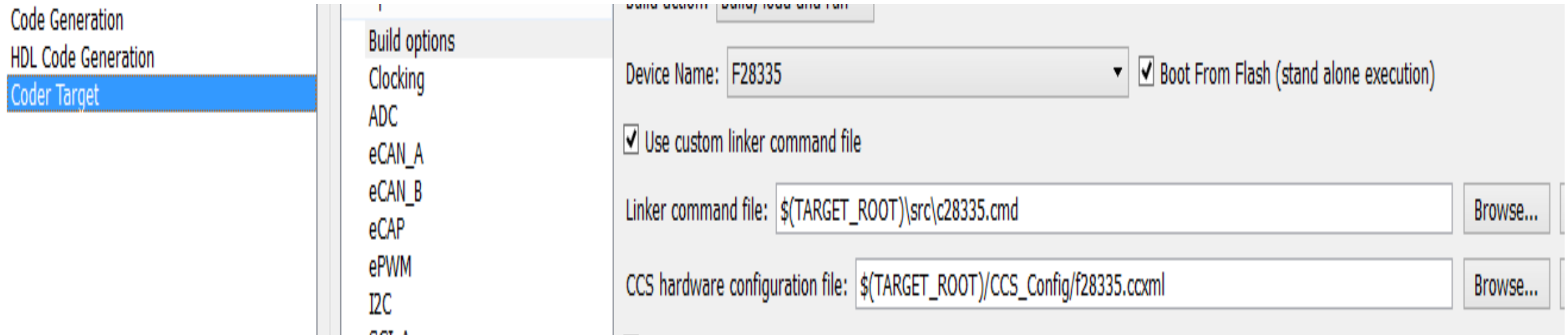
7- aller vers **Code Generation** et choisir le language C, puis sous Target Hardware spécifier TI Delfino F2833x dans



➤ Programmation de la C2000 F28335 avec MATLAB

- Installation des packages et configuration

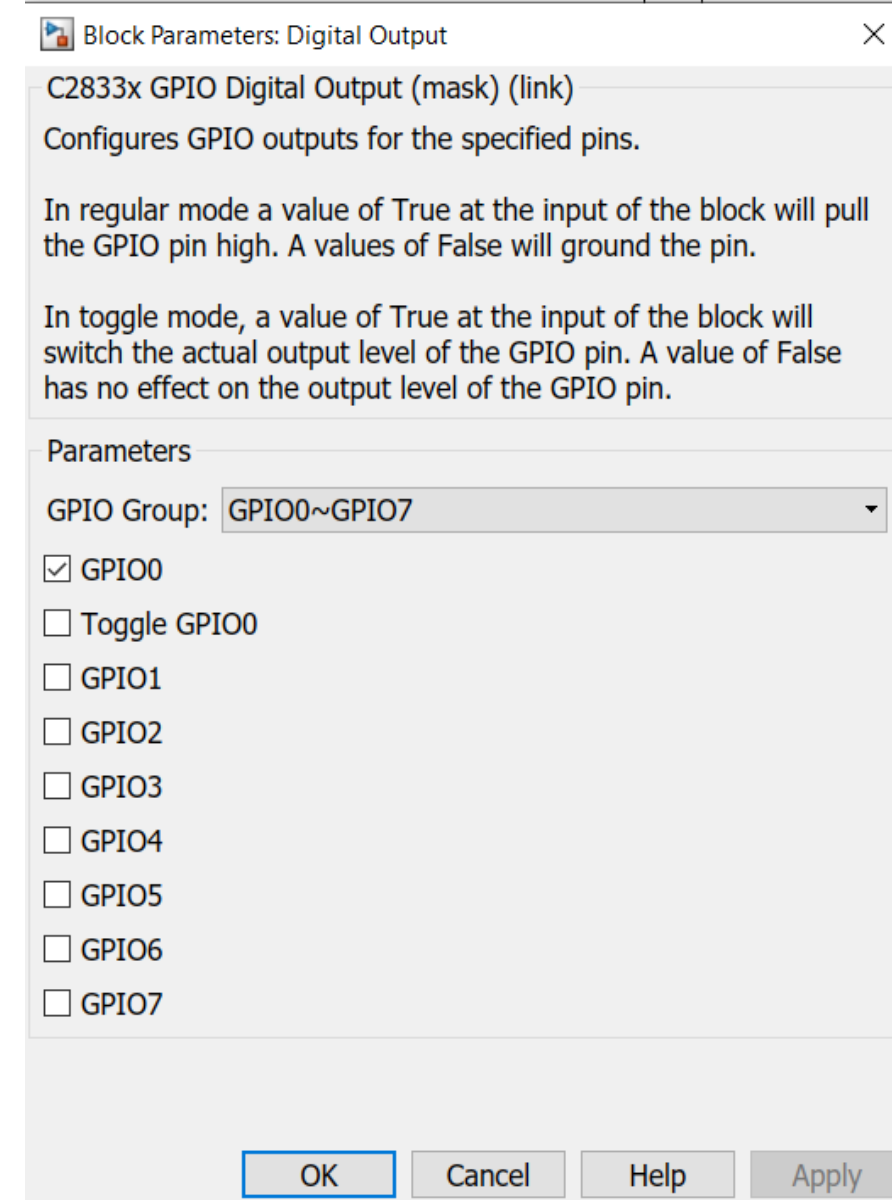
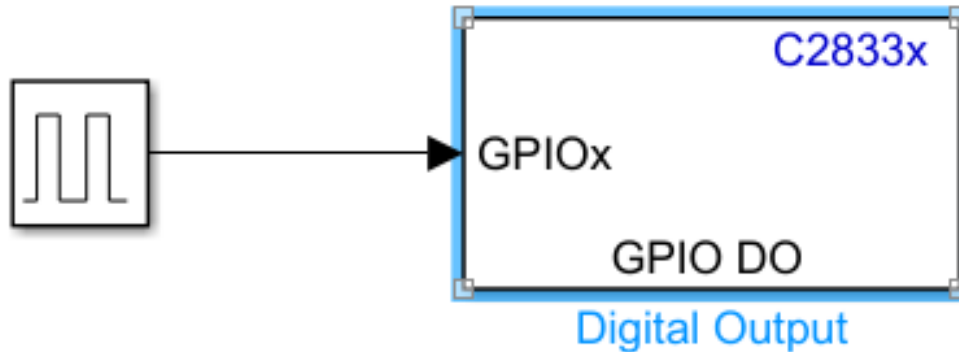
8- aller vers **Coder Target** et choisir, sous **Device Name**, choisir F28335



➤ Programmation de la C2000 F28335 avec MATLAB

■ Votre premier programme

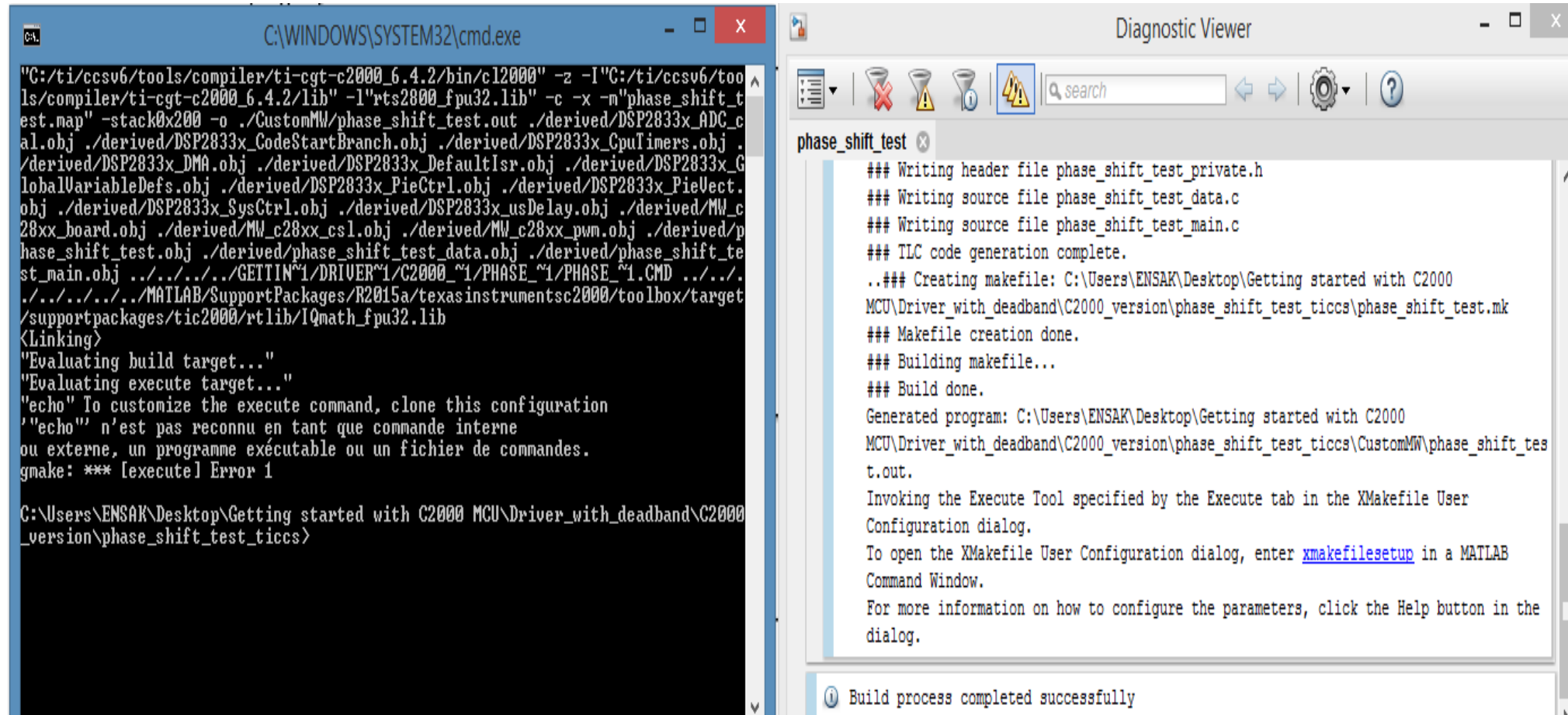
9- Maintenant que l'environnement Simulink est **correctement configuré**, vous pouvez démarrer un test pour garantir que le processus de construction réussit (**Building: Génération du code compatible avec la carte**). À cette fin, un modèle Simulink simple pourrait être créé comme le montre la figure suivante.



➤ Programmation de la C2000 F28335 avec MATLAB

■ Votre premier programme

10- Cliquer sur **Build** pour générer le code C (**file.out**) correspondant. Un message de succès doit s'afficher vers la fin de l'opération.



```
C:\WINDOWS\SYSTEM32\cmd.exe

"C:/ti/ccsv6/tools/compiler/ti-cgt-c2000_6.4.2/bin/cl2000" -z -I"C:/ti/ccsv6/too
ls/compiler/ti-cgt-c2000_6.4.2/lib" -l"rts2800_fpu32.lib" -c -x -m"phase_shift_t
est.map" -stack0x200 -o ./CustomMW/phase_shift_test.out ./derived/DSP2833x_ADC_c
al.obj ./derived/DSP2833x_CodeStartBranch.obj ./derived/DSP2833x_CpuTimers.obj .
./derived/DSP2833x_DMA.obj ./derived/DSP2833x_DefaultIsr.obj ./derived/DSP2833x_G
lobalVariableDefs.obj ./derived/DSP2833x_PieCtrl.obj ./derived/DSP2833x_PieVect.
obj ./derived/DSP2833x_SysCtrl.obj ./derived/DSP2833x_usDelay.obj ./derived/MW_c
28xx_board.obj ./derived/MW_c28xx_csl.obj ./derived/MW_c28xx_pwm.obj ./derived/p
hase_shift_test.obj ./derived/phase_shift_test_data.obj ./derived/phase_shift_te
st_main.obj ../../../../../../GETTING^1/DRIVER^1/C2000^1/PHASE_~1/PHASE_~1.CMD ../../
../../../../MATLAB/SupportPackages/R2015a/texasinstrumentsc2000/toolbox/target
/supportpackages/tic2000/rtlib/IQmath_fpu32.lib
<Linking>
"Evaluating build target..."
"Evaluating execute target..."
"echo" To customize the execute command, clone this configuration
"echo" n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
gmake: *** [execute] Error 1

C:\Users\ENSAK\Desktop\Getting started with C2000 MCU\Driver_with_deadband\C2000
_version\phase_shift_test_ticcs>
```

```
Diagnostic Viewer

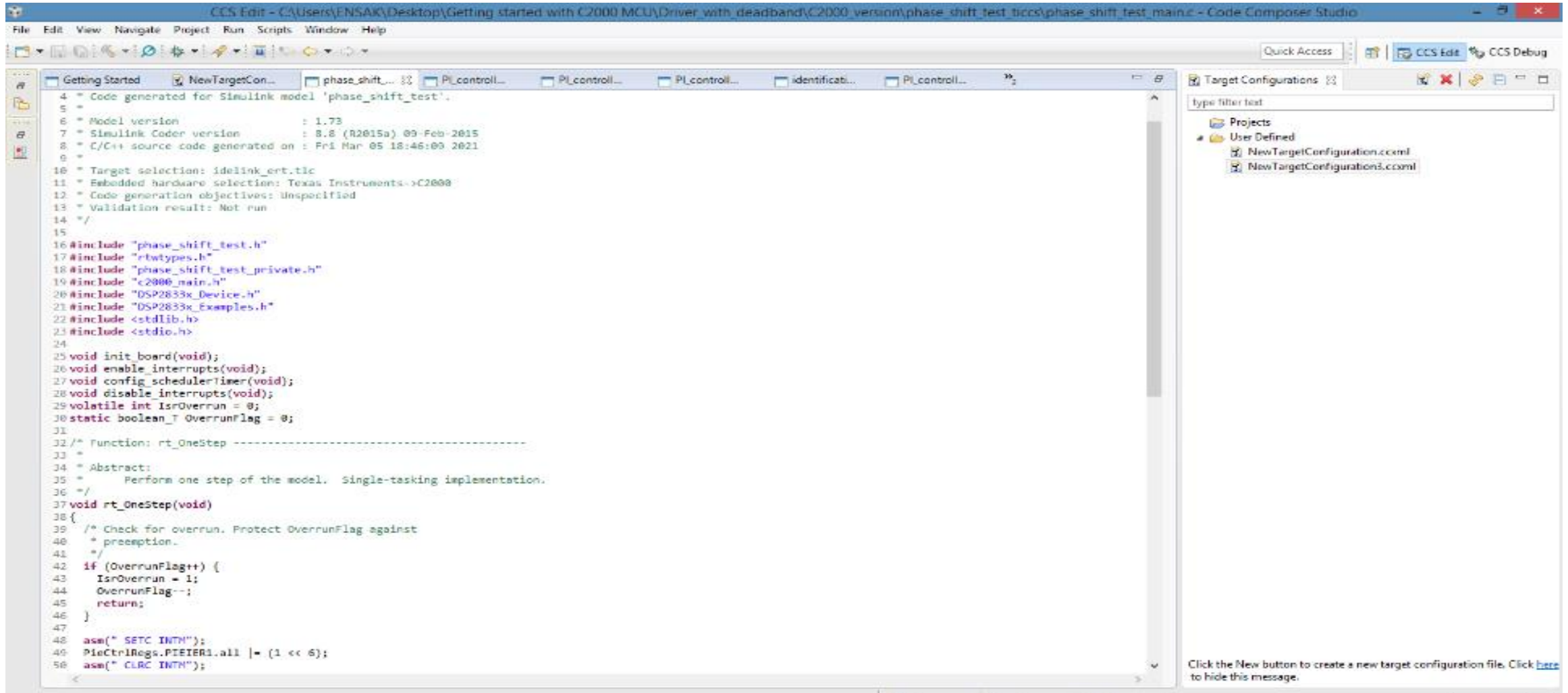
phase_shift_test x
  ### Writing header file phase_shift_test_private.h
  ### Writing source file phase_shift_test_data.c
  ### Writing source file phase_shift_test_main.c
  ### TLC code generation complete.
  ..### Creating makefile: C:\Users\ENSAK\Desktop\Getting started with C2000
MCU\Driver_with_deadband\C2000_version\phase_shift_test_ticcs\phase_shift_test.mk
  ### Makefile creation done.
  ### Building makefile...
  ### Build done.
  Generated program: C:\Users\ENSAK\Desktop\Getting started with C2000
MCU\Driver_with_deadband\C2000_version\phase_shift_test_ticcs\CustomMW\phase_shift_tes
t.out.
  Invoking the Execute Tool specified by the Execute tab in the XMakefile User
Configuration dialog.
  To open the XMakefile User Configuration dialog, enter xmakefilesetup in a MATLAB
Command Window.
  For more information on how to configure the parameters, click the Help button in the
dialog.

Build process completed successfully
```

➤ Programmation de la C2000 F28335 avec MATLAB

■ Votre premier programme

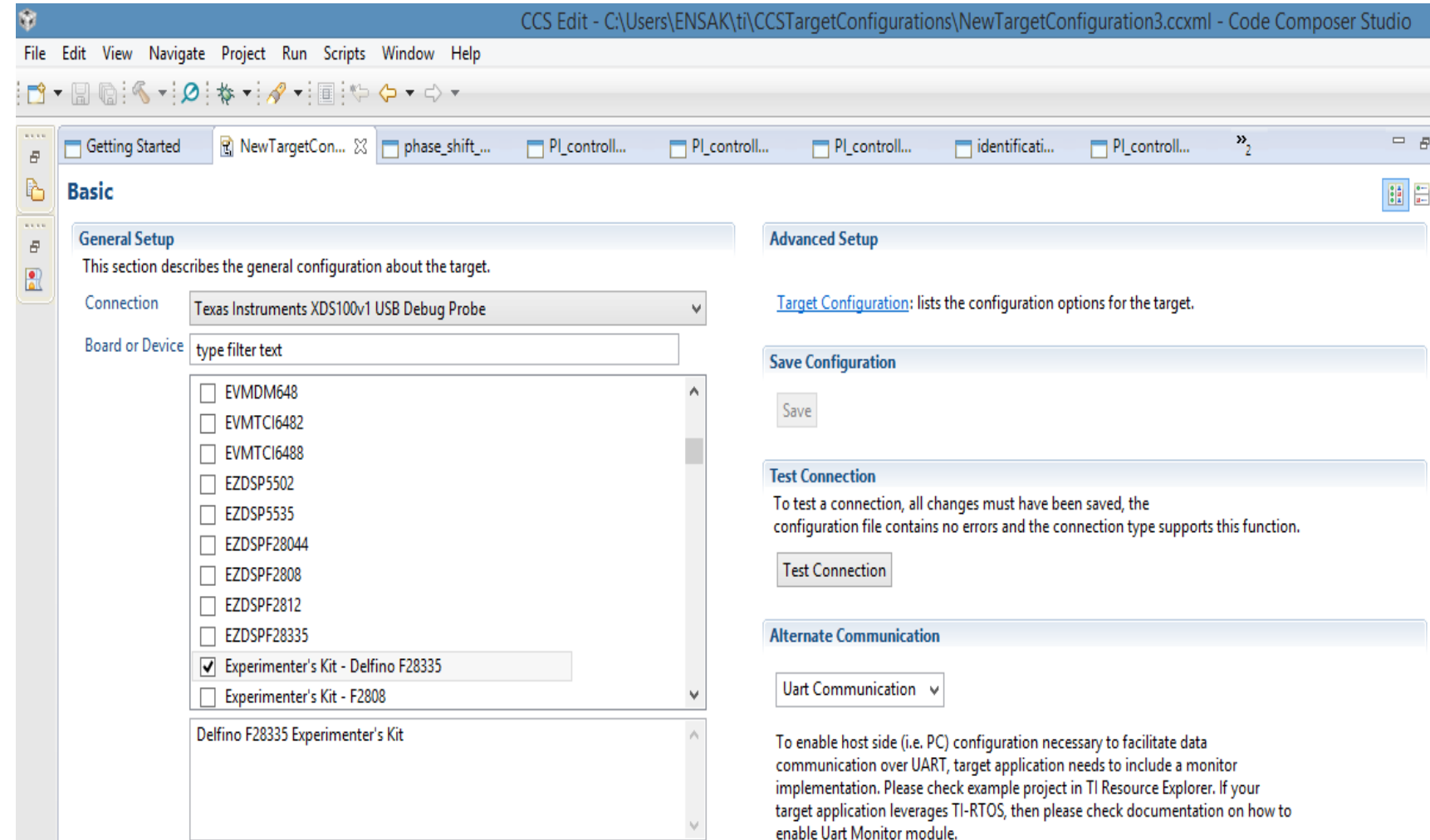
11- Lancer Code Composer Studio et ouvrez le fichier généré



➤ Programmation de la C2000 F28335 avec MATLAB

■ Votre premier programme

12- choisir **File New Targer Configuration** et puis choisir **Experimenter's Kit Delfino F28335**

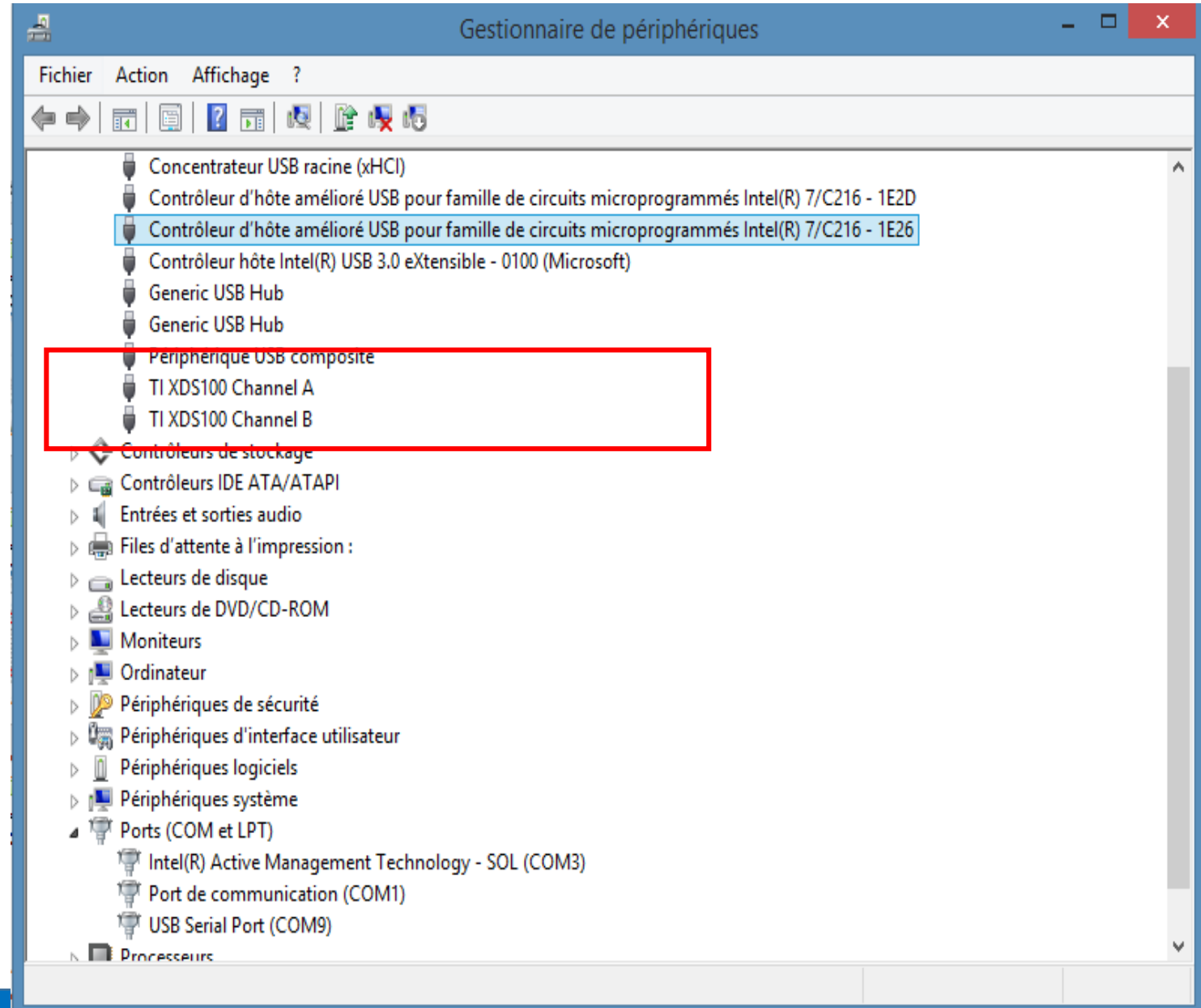


➤ Programmation de la C2000 F28335 avec MATLAB

■ Votre premier programme

13- À ce stade, connectez la carte à l'ordinateur hôte (PC) via un câble USB et assurez-vous que **les pilotes** requis sont correctement installés. (voir figure ci-dessous)

14- Cliquer sur Load File pour programmer la carte et tester le fonctionnement du code

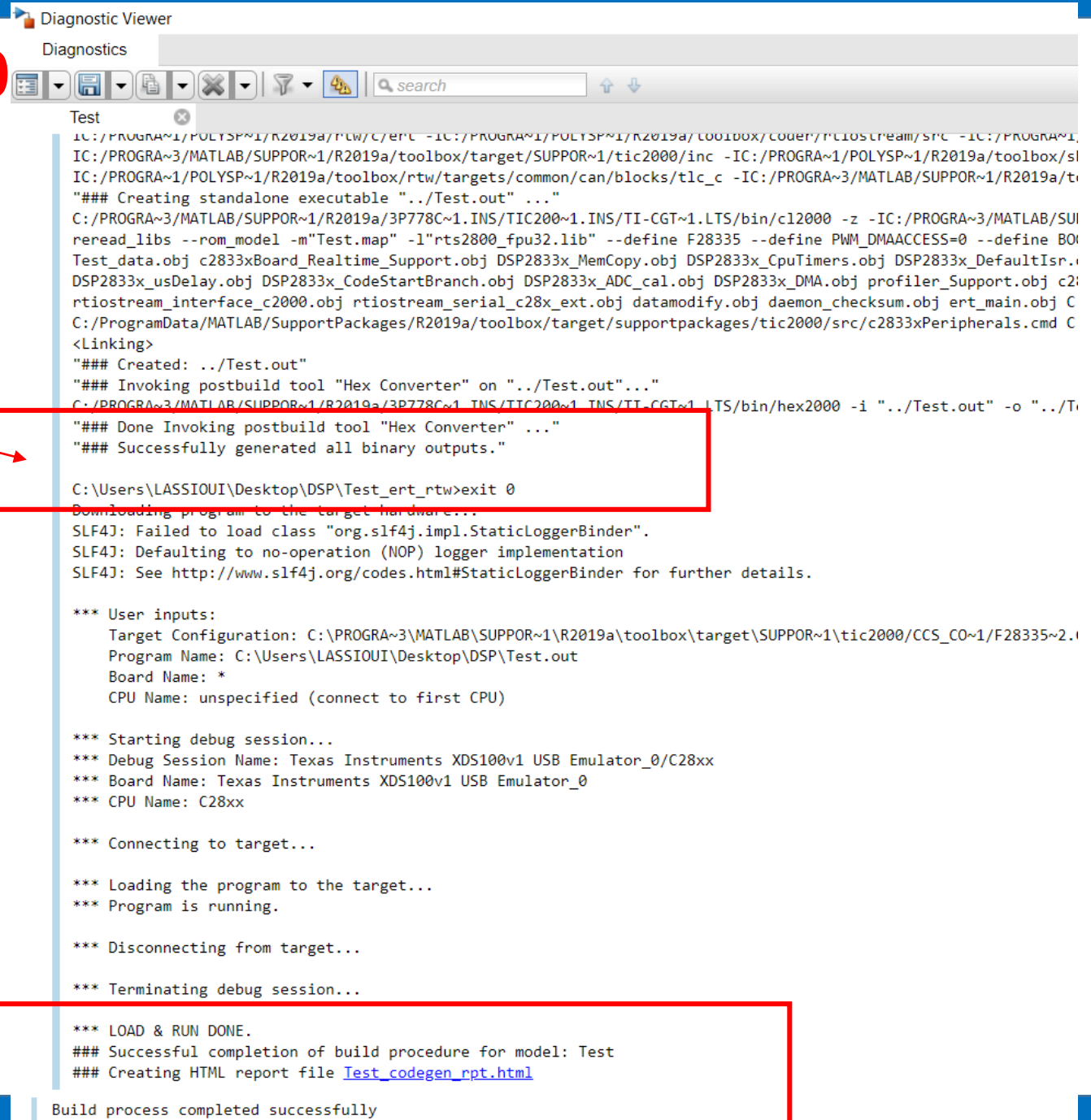


➤ Programmation de la C2000

■ Votre premier programme

Ce message indique que le compilateur a bien généré tous les fichiers nécessaires pour la programmation du DSP

Ce message indique que le programme a été chargé dans la mémoire du DSP ainsi son exécution a commencé



```
Test
IC:/PROGRA~1/POLYSP~1/R2019a/rtw/c/ert -IC:/PROGRA~1/POLYSP~1/R2019a/toolbox/coder/rtiostream/src -IC:/PROGRA~1
IC:/PROGRA~3/MATLAB/SUPPOR~1/R2019a/toolbox/target/SUPPOR~1/tic2000/inc -IC:/PROGRA~1/POLYSP~1/R2019a/toolbox/sl
IC:/PROGRA~1/POLYSP~1/R2019a/toolbox/rtw/targets/common/can/blocks/tlc_c -IC:/PROGRA~3/MATLAB/SUPPOR~1/R2019a/t
"### Creating standalone executable "../Test.out" ..."
C:/PROGRA~3/MATLAB/SUPPOR~1/R2019a/3P778C~1.INS/TIC200~1.INS/TI-CGT~1.LTS/bin/cl2000 -z -IC:/PROGRA~3/MATLAB/SUI
reread_libs --rom_model -m"Test.map" -l"rts2800_fpu32.lib" --define F28335 --define PWM_DMAACCESS=0 --define BO
Test_data.obj c2833xBoard_Realtime_Support.obj DSP2833x_MemCopy.obj DSP2833x_CpuTimers.obj DSP2833x_DefaultIsr.o
DSP2833x_usDelay.obj DSP2833x_CodeStartBranch.obj DSP2833x_ADC_cal.obj DSP2833x_DMA.obj profiler_Support.obj c2
rtiostream_interface_c2000.obj rtiostream_serial_c28x_ext.obj datamodify.obj daemon_checksum.obj ert_main.obj C
C:/ProgramData/MATLAB/SupportPackages/R2019a/toolbox/target/supportpackages/tic2000/src/c2833xPeripherals.cmd C
<Linking>
"### Created: ../Test.out"
"### Invoking postbuild tool "Hex Converter" on "../Test.out"..."
C:/PROGRA~3/MATLAB/SUPPOR~1/R2019a/3P778C~1.INS/TIC200~1.INS/TI-CGT~1.LTS/bin/hex2000 -i "../Test.out" -o "../T
"### Done Invoking postbuild tool "Hex Converter" ..."
"### Successfully generated all binary outputs."

C:\Users\LASSIOUI\Desktop\DSP\Test_ert_rtw>exit 0
Downloading program to the target hardware...
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

*** User inputs:
Target Configuration: C:\PROGRA~3\MATLAB\SUPPOR~1\R2019a\toolbox\target\SUPPOR~1\tic2000\CCS_CO~1\F28335~2.1
Program Name: C:\Users\LASSIOUI\Desktop\DSP\Test.out
Board Name: *
CPU Name: unspecified (connect to first CPU)

*** Starting debug session...
*** Debug Session Name: Texas Instruments XDS100v1 USB Emulator_0/C28xx
*** Board Name: Texas Instruments XDS100v1 USB Emulator_0
*** CPU Name: C28xx

*** Connecting to target...

*** Loading the program to the target...
*** Program is running.

*** Disconnecting from target...

*** Terminating debug session...

*** LOAD & RUN DONE.
### Successful completion of build procedure for model: Test
### Creating HTML report file Test\_codegen\_rpt.html

Build process completed successfully
```