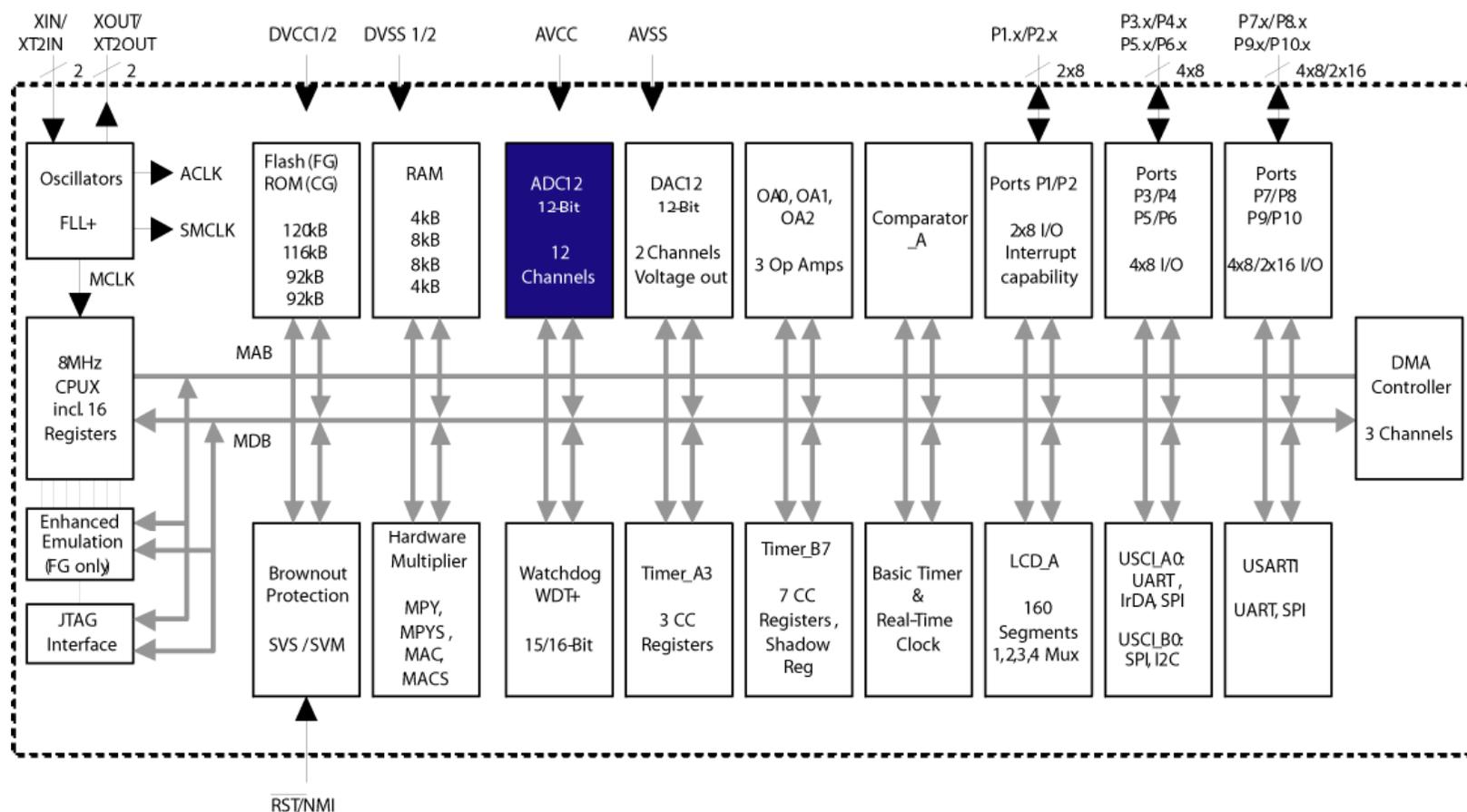


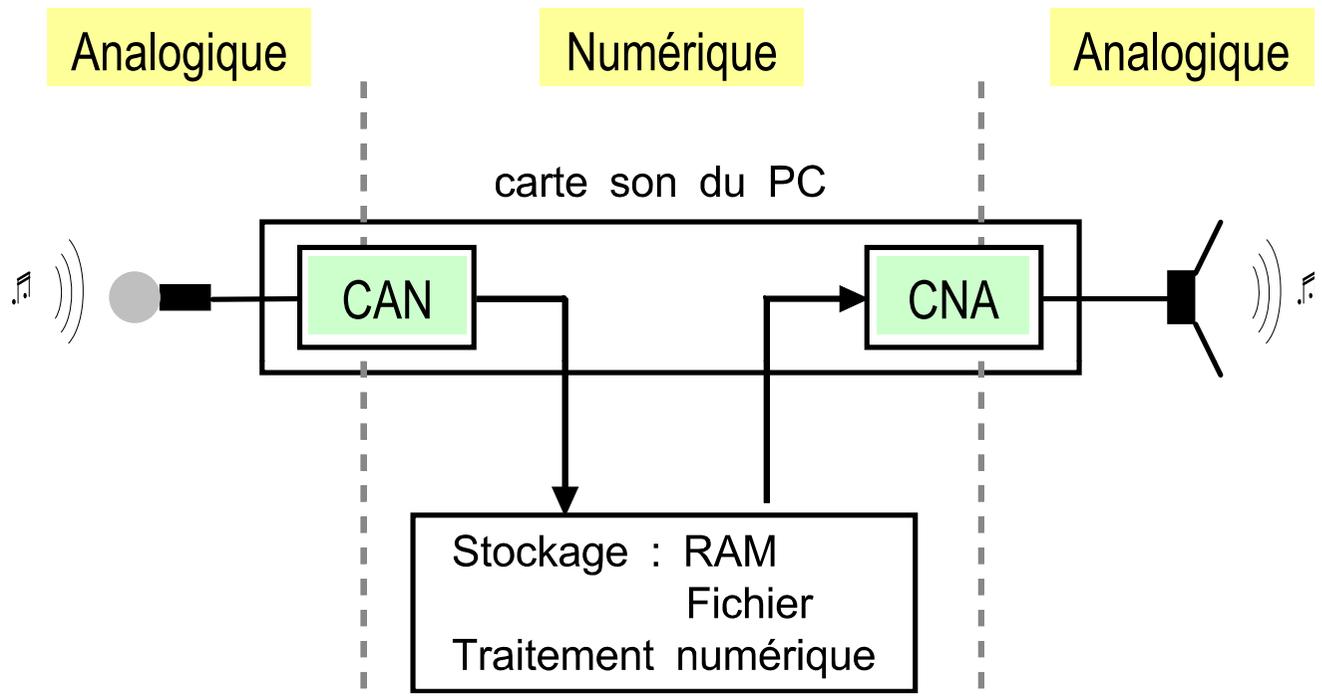
Conversion analogique-numérique



Convertisseur Analogique-Numérique

Un Convertisseur Analogique-Numérique (CAN en français, *ADC* pour *Analog to Digital Converter* en anglais) est un circuit dont la fonction est de générer à partir d'une valeur analogique, une valeur numérique (codée sur plusieurs bits), proportionnelle à la valeur analogique entrée.

Exemple d'un enregistrement sonore

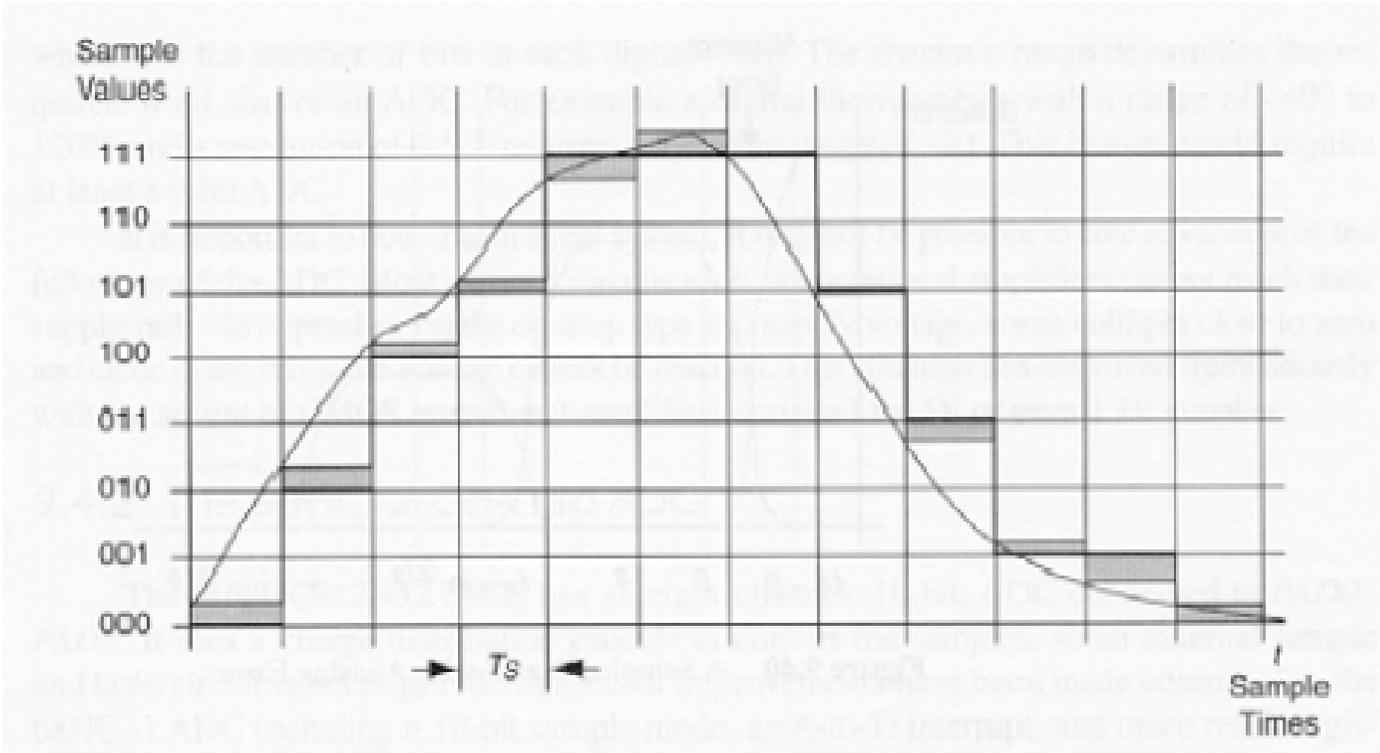


Convertisseur **A**nalogique **N**umérique : *Analog to Digital Converter*
🇫🇷 CAN : ADC 🇬🇧

Convertisseur **N**umérique **A**nalogique : *Digital to Analog Converter*
🇫🇷 CNA : DAC 🇬🇧

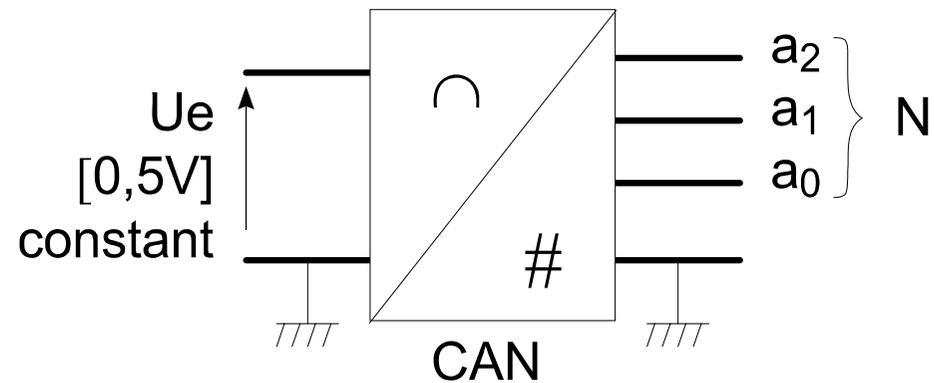
Conversion Analogique-Numérique


La conversion Analogique-Numérique introduit toujours une erreur de quantification

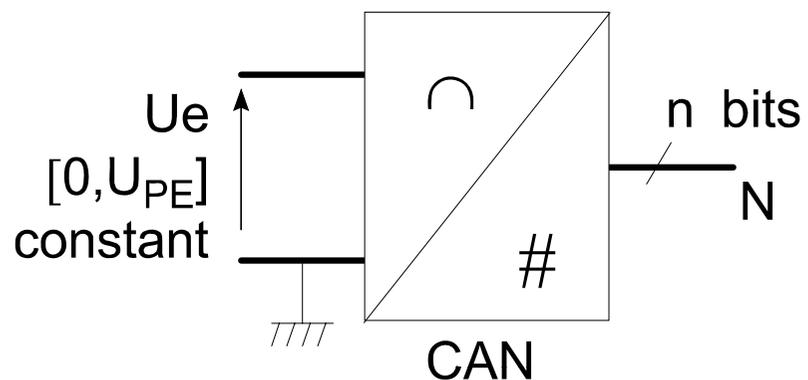


Conversion Analogique-Numérique

- Exemple d'un CAN 3 bits ($n = 3$)

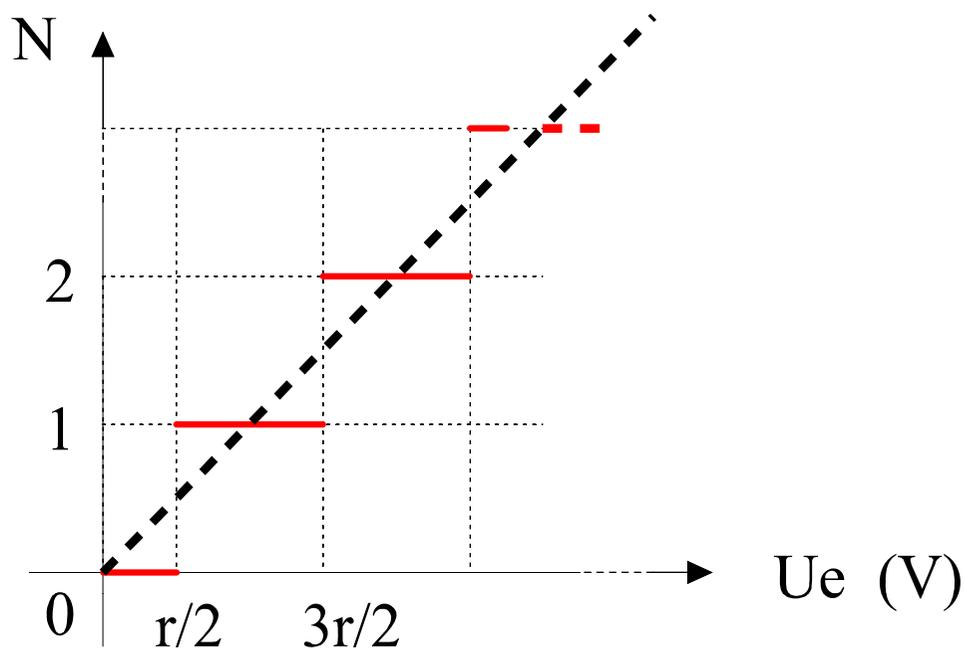


Erreur de quantification



Résolution analogique : $r = U_{PE}/2^n$

Résolution numérique : n bits



Erreur analogique : $\pm r/2$

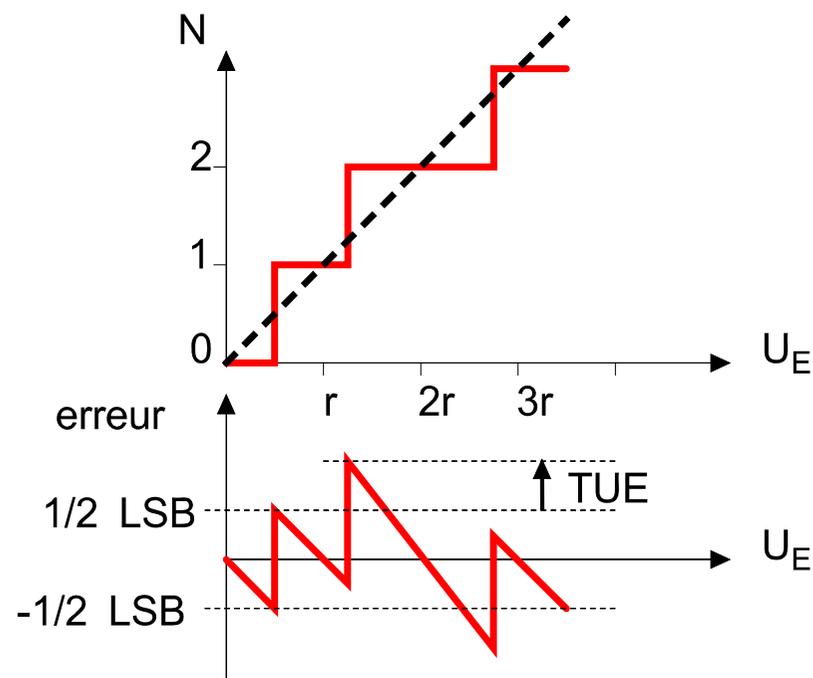
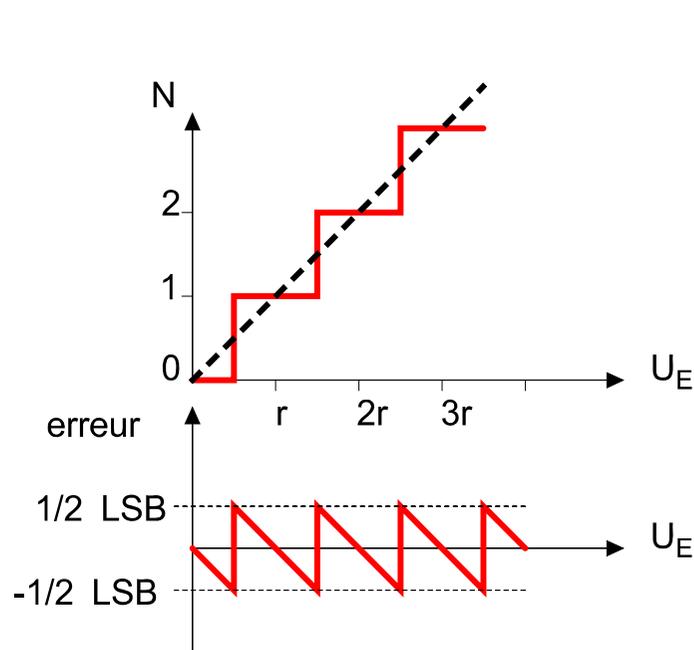
Erreur numérique : $\pm 1/2$ LSB

Autres erreurs

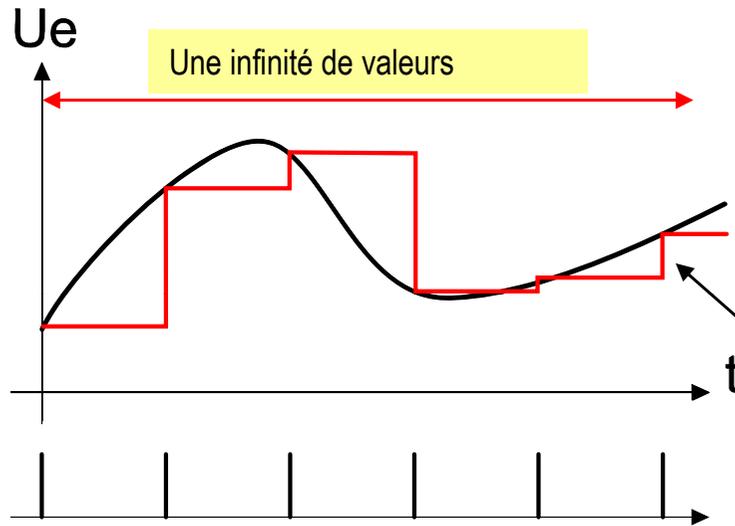
À l'erreur de quantification, s'ajoutent d'autres erreurs: linéarité, offset, gain ...

Le constructeur fournit en général la valeur max

TUE : *Total Unadjusted Error*



Conversion d'une tension variable



En amont du CAN se trouve un échantillonneur-bloqueur qui prélève régulièrement une valeur de U_e et bloque cette valeur jusqu'à l'échantillon suivant. (mémoire analogique)

Tension d'entrée du CAN
le temps de conversion doit être inférieur à T_e

Commande de l'échantillonneur
Période T_e , fréquence $F_e = 1/T_e$

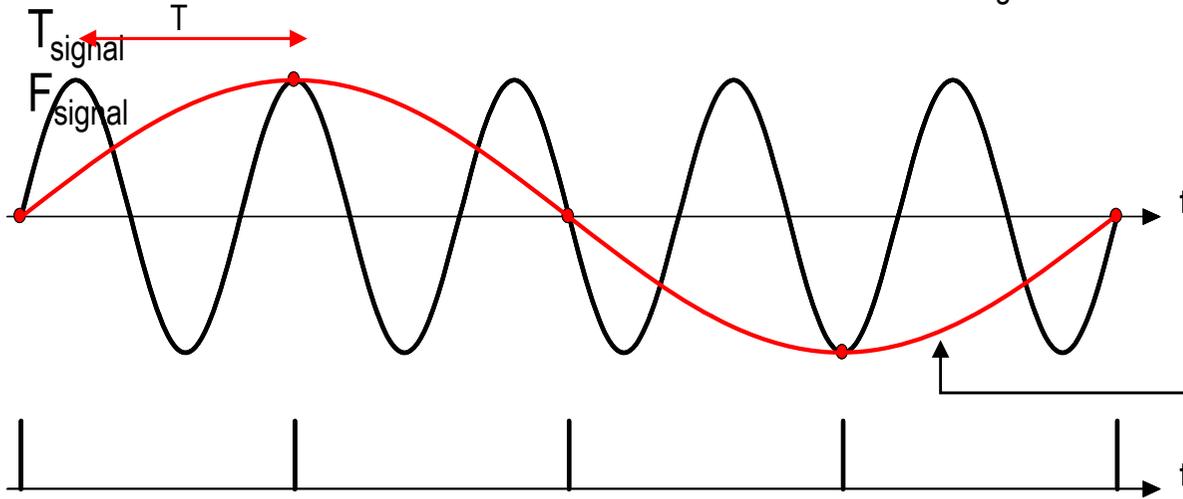
La conversion analogique numérique implique une double quantification :

- quantification temporelle (échantillonnage)
- quantification en amplitude (résolution)

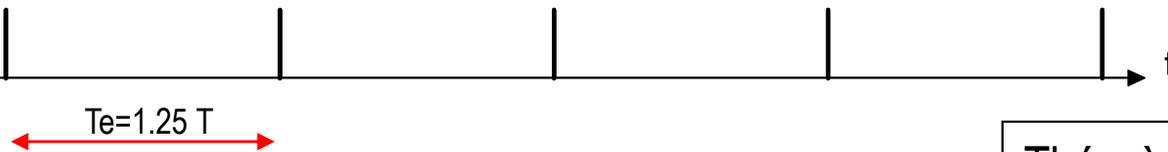
Repliement du spectre ou crènelage (*aliasing*)

Exemple avec un signal sinusoïdal de période T_{signal} échantillonné à $T_e = 1.25$

$F_e = 0.8$

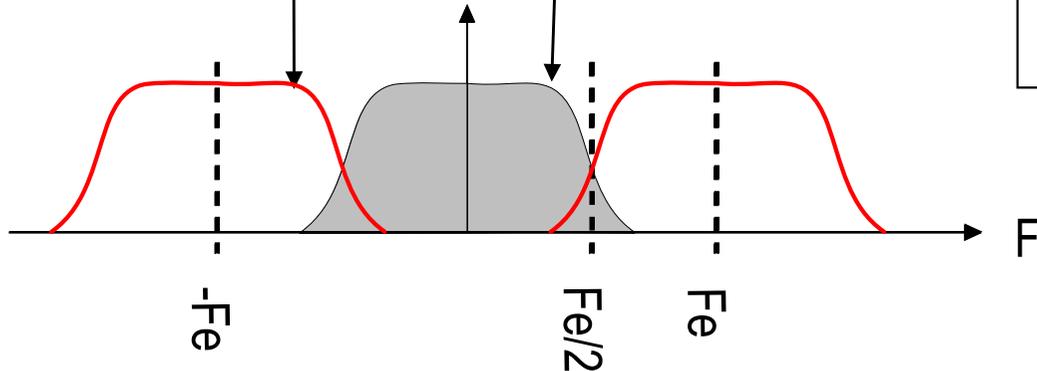


$T' = 5 T$
 $F' = 0.2 F$
 $F' = F - F_e$



Spectre du signal à échantillonner

Repliement du spectre

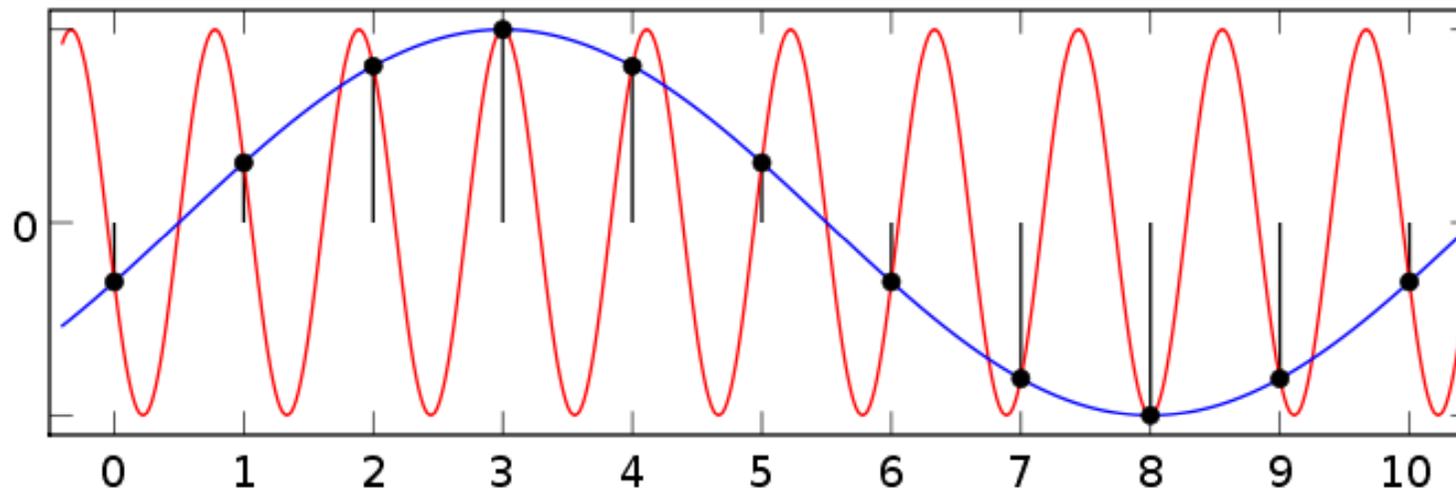


Théorème de SHANNON:
 (Critère de Nyquist)
 $F_e > 2 \cdot F_{\text{max}}$
 F_{max} : fréquence supérieure du spectre de U_e

À l'entrée d'un CAN/ADC il faut donc un filtre passe bas qui coupe à $F_c = F_e/2$ mais évidemment on perd l'information des fréquences supérieurs ...

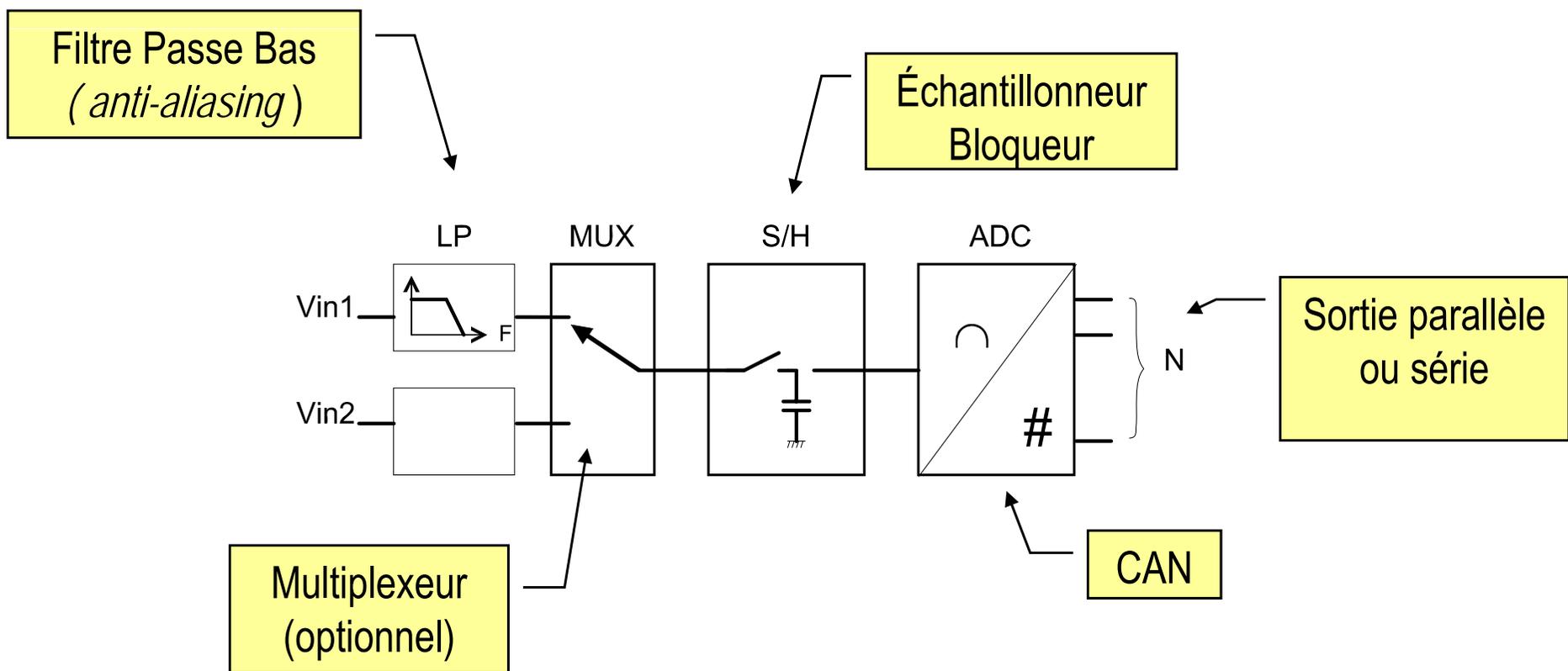
Aliasing ou repliement

- Le repliement consiste en le « repli » d'une fréquence supérieure à la fréquence de Nyquist vers une fréquence inférieure à celle-ci.
- Par exemple dans le cas d'un signal sonore, ceci est particulièrement gênant puisque ce phénomène peut transformer une harmonique d'un son en une fréquence anharmonique, déplaisante à l'oreille.



Cette graphique démontre le repliement d'un [signal sinusoïdal](#) de fréquence $f = 0.9$, qui est interprété comme un signal de fréquence $f = 0.1$ lors d'un [échantillonnage](#) de période $T = 1.0$.

Configuration typique



Technologies de conversion Analogique-Numérique

Il existe plusieurs solutions pour convertir un signal analogique en signal numérique elles sont classées ici dans l'ordre de la moins rapide à la plus rapide.

Technologie	Temps de conversion	Exemple d'utilisation
Simple rampe	Lent (ms)	Mesure sans précision
Double rampe Multi rampe	Lent (ms)	Multimètre
Approximations successives	Rapide (μ s)	Acquisition son
Flash (ou CAN parallèle)	Très rapide (ns)	Acquisition vidéo Oscilloscope numérique

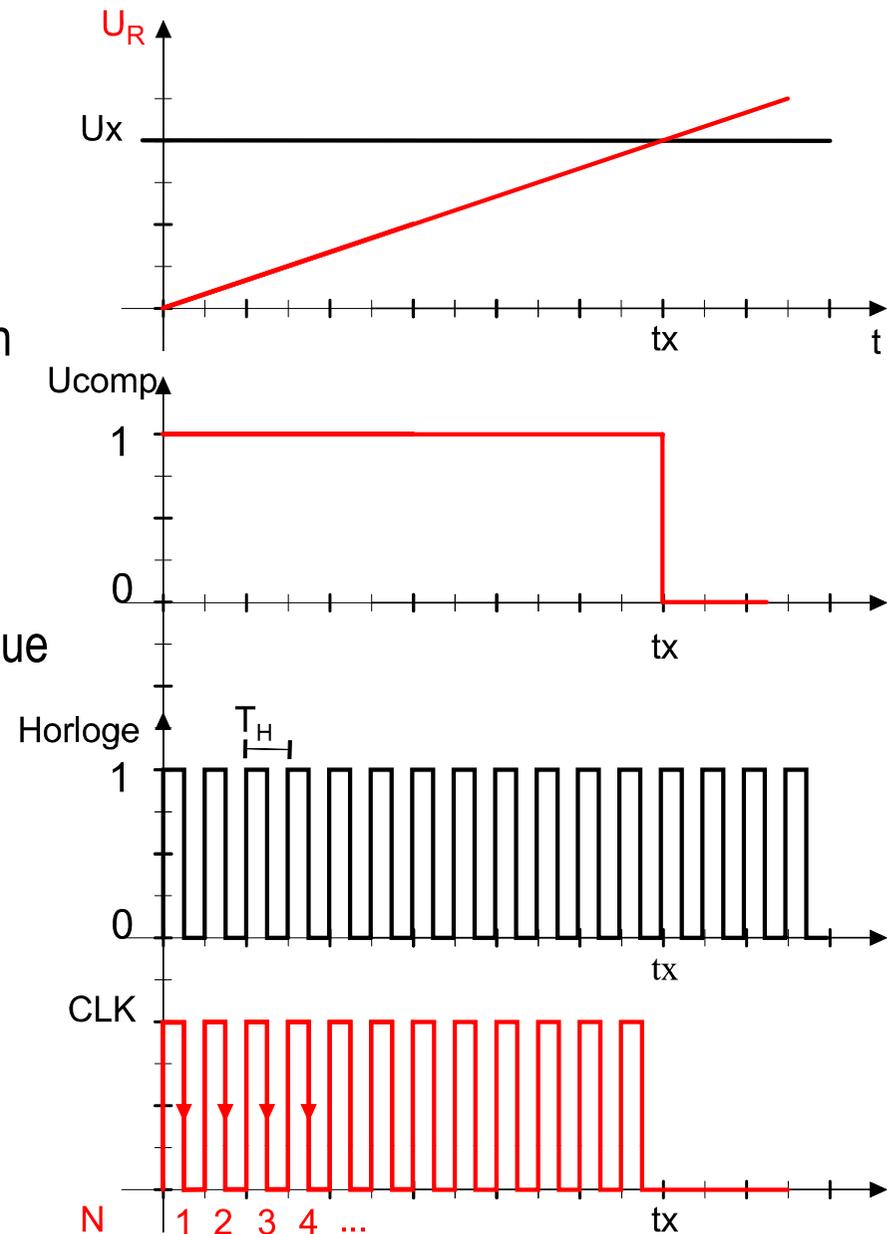
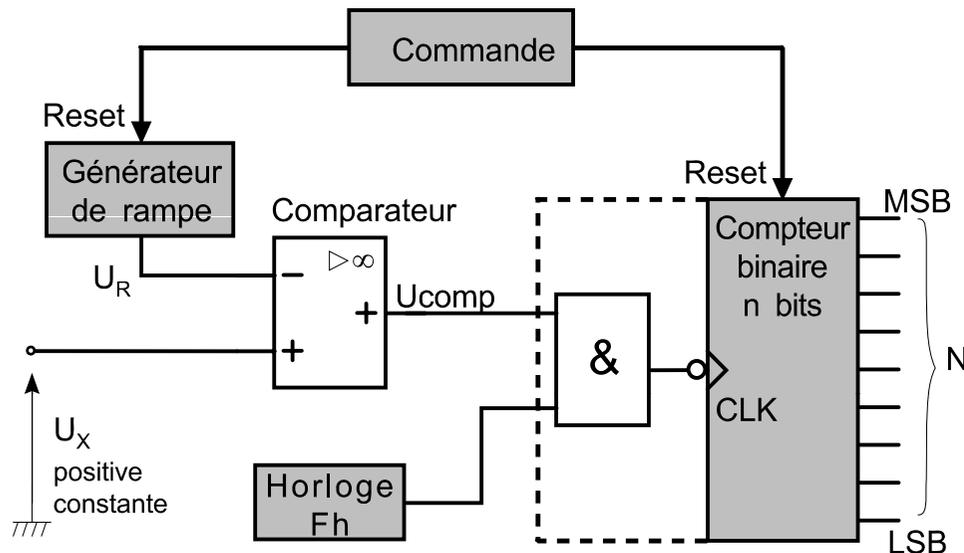
Autres technologies :

- CAN pipeline
- CAN Sigma-Delta ($\Sigma-\delta$) à sur-échantillonnage

Périphérique : le convertisseur AD

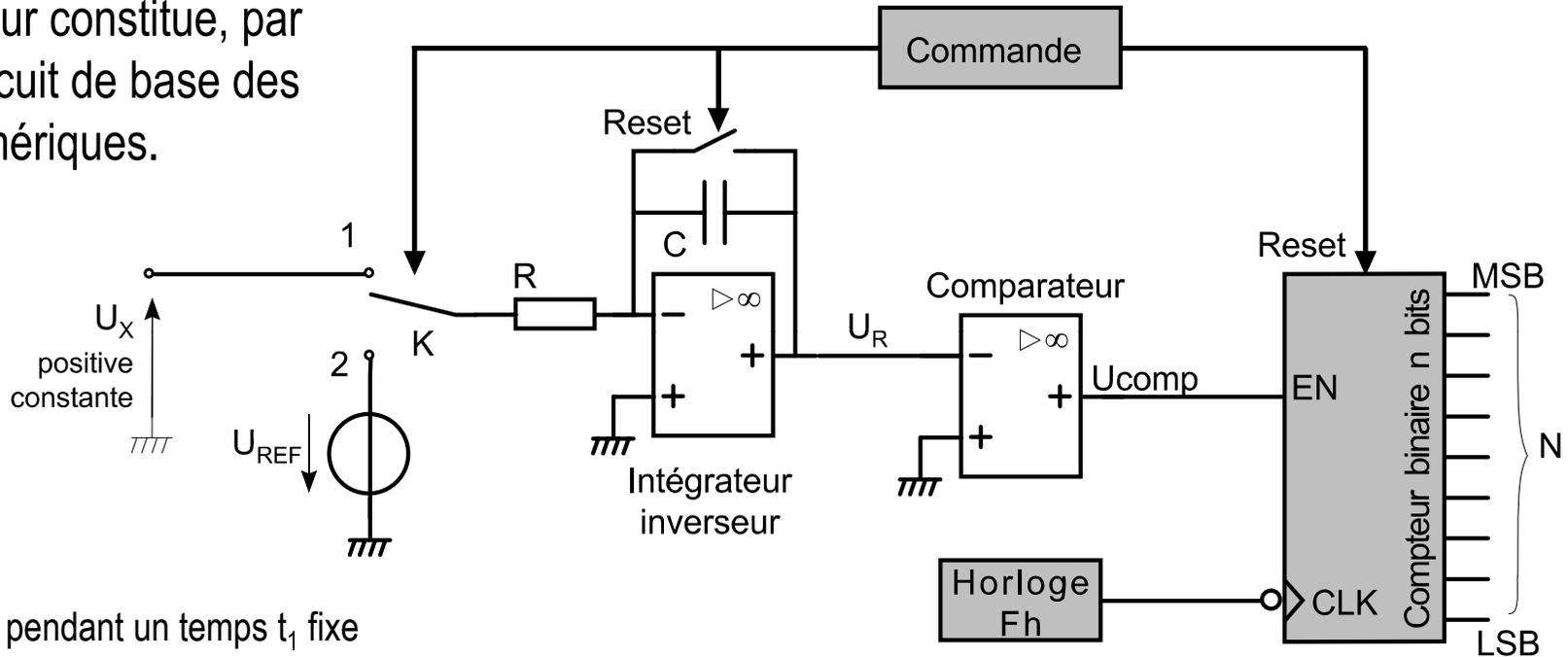
Convertisseur à simple rampe

- On réalise au moyen d'un compteur et d'un convertisseur numérique-analogique une rampe de tension.
- Un comparateur arrête le compteur lorsque la tension créée par le CNA atteint la tension à convertir.
- Le compteur indique alors le résultat sur N bits, qui peut être stocké ou traité.
- Un inconvénient est que le temps de conversion évolue avec la tension à convertir.



CAN double rampe

Ce convertisseur constitue, par exemple, le circuit de base des voltmètres numériques.



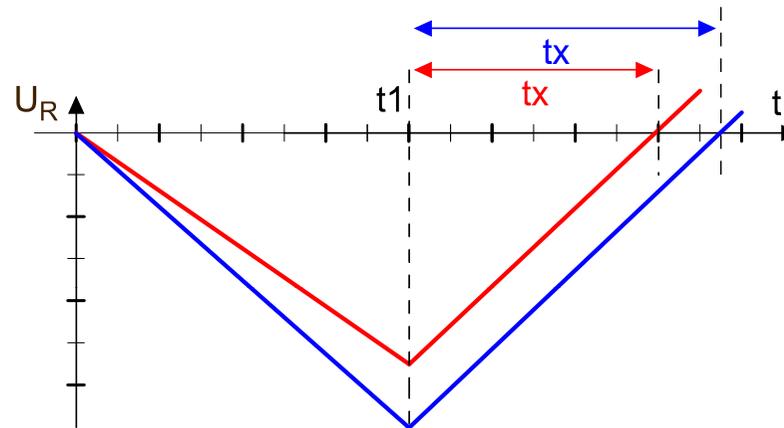
1. On intègre U_x pendant un temps t_1 fixe
2. On intègre $-U_{REF}$
On mesure le temps t_x (en unité T_H) que met U_R pour revenir à 0

$$U_{Rmax} = - U_x \cdot t_1 / RC$$

$$U_{Rmax} = - U_{REF} \cdot t_x / RC$$

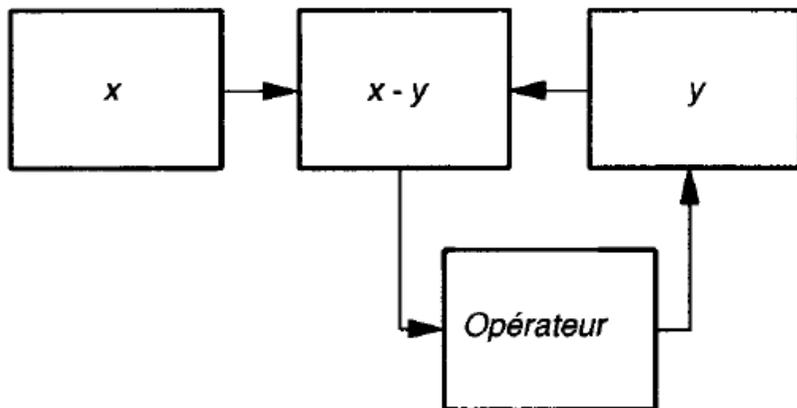
$$t_x = t_1 \cdot U_x / U_{REF}$$

$$N = t_x / T_H = (t_1 / T_H) \cdot (U_x / U_{REF})$$



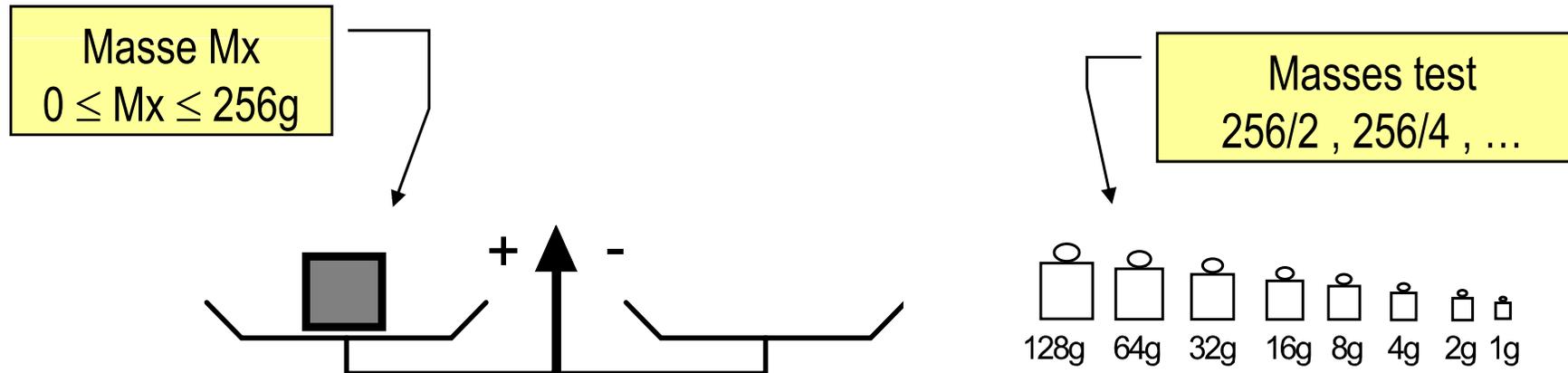
Convertisseurs à approximations successives

Ici la conversion Analogique-Numérique est essentiellement une mesure par comparaison.



Principe d'une conversion (mesure) par approximations successives

Principe de dichotomie : on divise la plage de recherche par 2 à chaque étape :



1er test : on compare Mx et 128g (le poids fort)

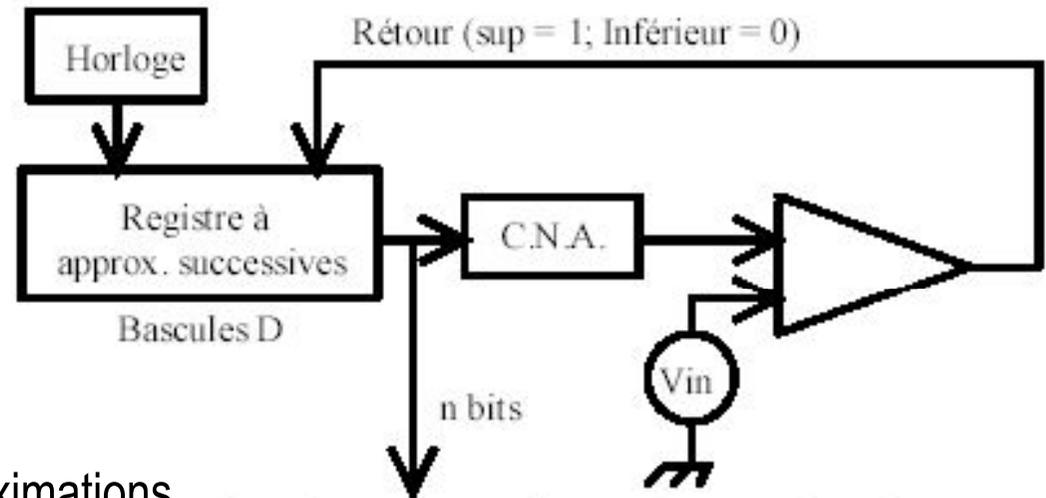
- : $Mx < 128g$: on enlève la masse de 128g

+ : $Mx > 128g$: on conserve la masse de 128g

2ème test : on ajoute 64g ...

On réalise dans cet exemple une mesure de Mx en 8 tests
avec une résolution de 1g .

Convertisseur à approximations successives par transfert de charge



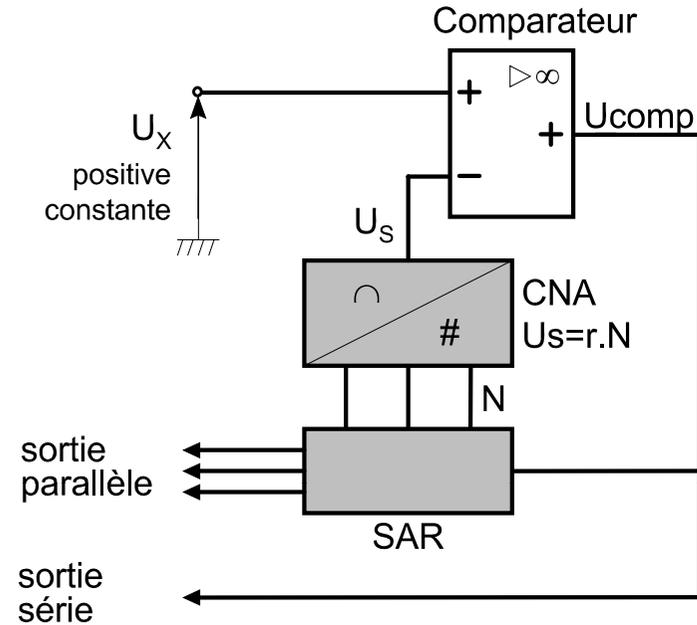
- Ce sont les plus courants des CAN à approximations successives, ils utilisent des transferts de charge dans un réseau de condensateurs pondérés.
- Le "cerveau" de ces CAN est un registre séquenceur:
SAR = Successive Approximation Register
- Le séquenceur SAR, couplé à un convertisseur Numérique-Analogique (CNA/DAC) génère une tension analogique, qui est comparée au signal à convertir.
- Le résultat de cette comparaison est alors introduit dans le SAR, qui va le prendre en compte, pour la suite du processus de dichotomie, jusqu'à complétion.
- Le convertisseur réalise donc sa conversion en positionnant en premier le bit de poids fort (MSB) et en descendant progressivement jusqu'au LSB.

Exemple d'un CAN 3 bits

La sortie du CNA est une tension analogique $U_s = r \cdot N$

On teste successivement les bits de N en débutant par le poids fort (MSB – Most Significant Bit)
Le résultat du test est donné par le comparateur.

Exemple avec $r=1V$, $U_{PE}=8V$, $U_x=4.5V$



N	$U_s=r.N$	U_{comp}	
100b=4	4V	1	Test du MSB: $4.5 > 4$ on garde MSB à 1 ...
110b=6	6V	0	et on teste le bit suivant: $4.5 < 6$ on remet le bit à 0
101b=5	5V	0	$4.5 < 5$, le LSB = 0 , le nombre cherché est 100b

Sortie série
(poids fort en 1^{er})

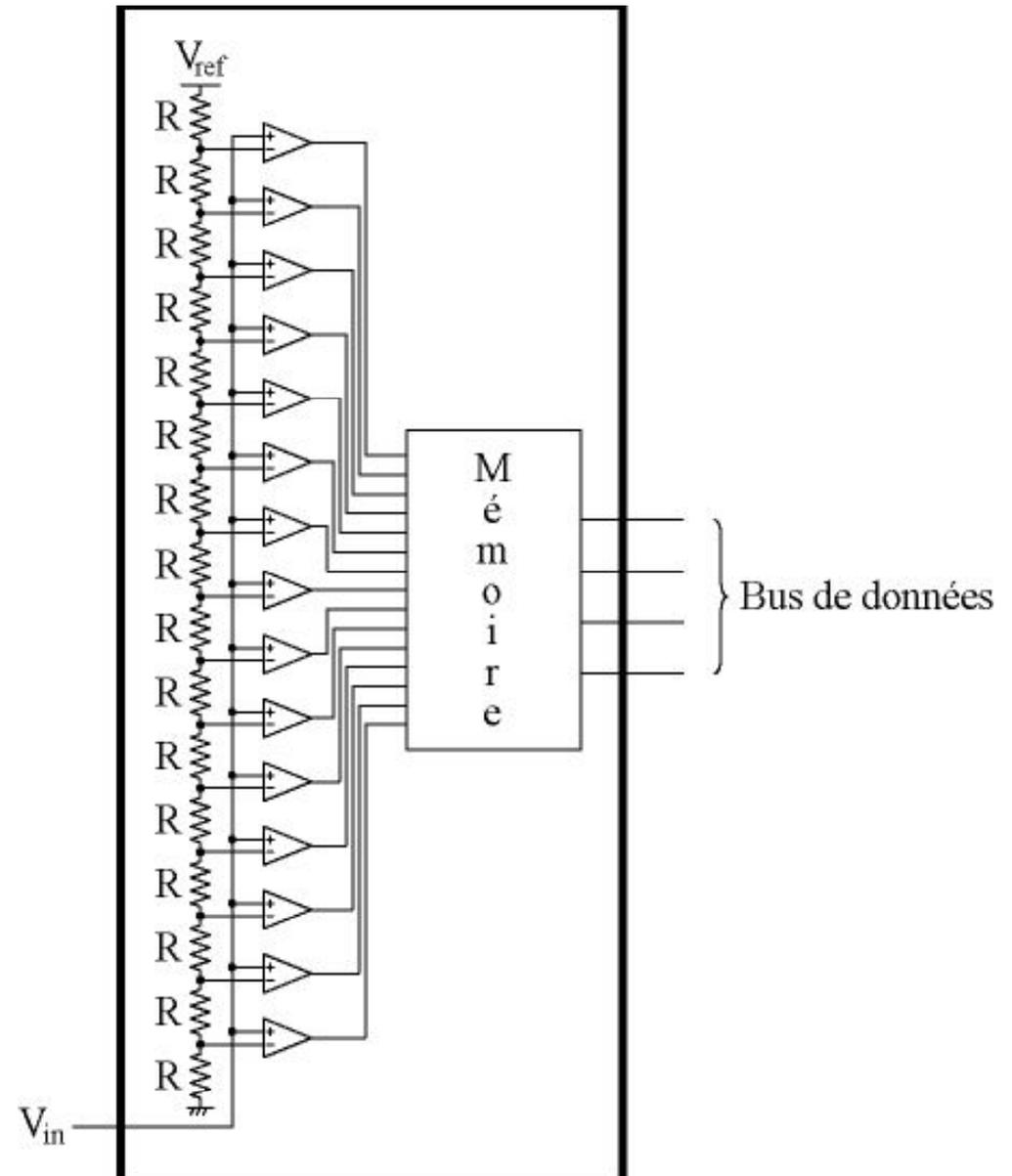
Sortie parallèle : 100b

Pour un CAN de n bits il faudra n tests

Les convertisseurs à approximations successives ont des temps de conversion de l'ordre de la dizaine de microsecondes, pour des résolutions d'une douzaine de bits environ.

Convertisseur Flash

- Le principe est de générer $(2N - 1)$ tensions analogiques au moyen d'un diviseur de tension à $2N$ résistances.
- Les $(2N - 1)$ tensions obtenues aux bornes de chacune des résistances sont ensuite comparées dans $(2N - 1)$ comparateurs au signal à convertir.
- Un bloc logique combinatoire relié à ces comparateurs donnera le résultat codé sur N bits en parallèle.
- Cette technique de conversion est très rapide, mais coûteuse en composants et donc utilisée pour les applications critiques comme la vidéo.
- Les convertisseurs Flash ont des temps de conversion inférieurs à la microseconde mais une précision assez faible (de l'ordre de la dizaine de bits).



Caractéristiques principales de l'ADC du MSP430

Convertisseur ADxx (10, 12 ou 16) bits à approximations successives

Jusqu'à 200 kSPS: 200'000 échantillons par seconde, ou une période de 5 μ s

Contrôle du démarrage des conversions par software ou par Timer

Plusieurs canaux d'entrée configurable individuellement

4 canaux supplémentaires (Température interne, V_{CC} , V_{eREF+} , V_{eREF-})

Contrôle des séquences d'acquisition (simple, répétitive, unique, séquentielle)

Contrôle de mise en veille

Buffer de stockage des entrées analogiques

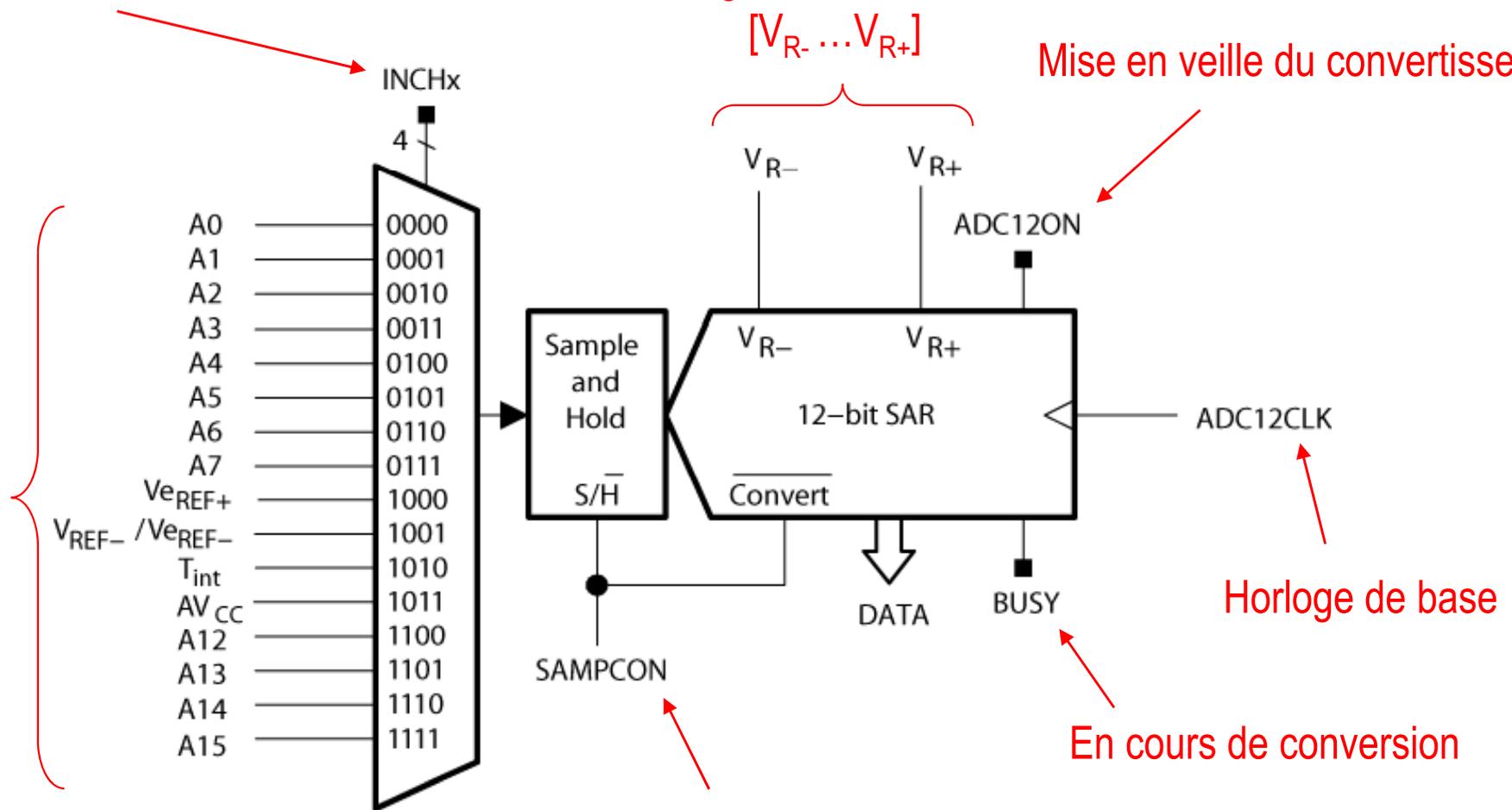
Cœur du convertisseur A/D

Sélection du canal à convertir

Plage de conversion
 $[V_{R-} \dots V_{R+}]$

Mise en veille du convertisseur

16 canaux analogiques



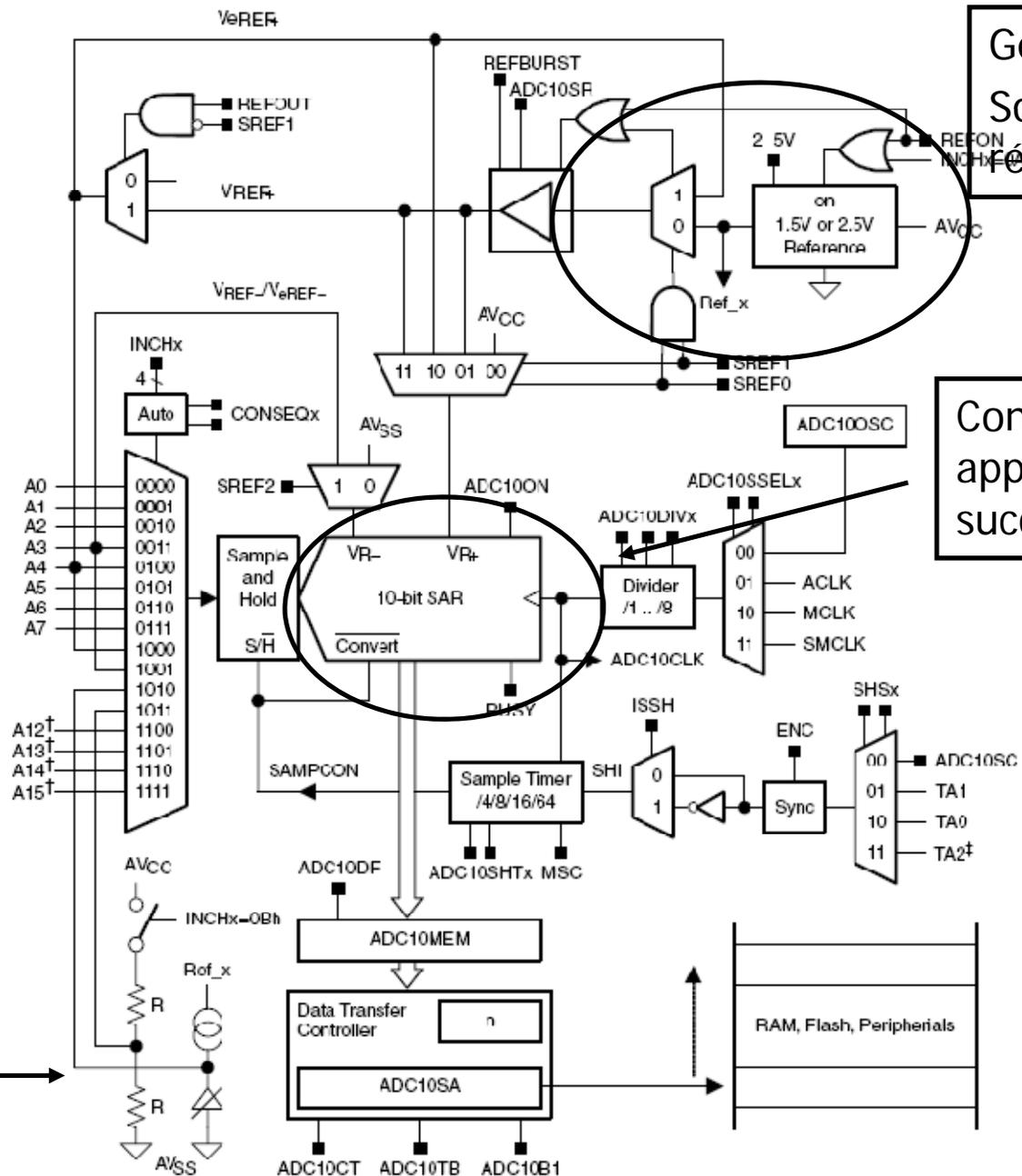
Contrôle du S&H et du démarrage de la conversion 24

Générateur
Source de tension
référence

Convertisseur à
approximations
successives

8 entrées
analogiques

Capteur Temp°
intégré



†MSP430x22xx devices only. Channels A12-A15 tied to channel A11 in other devices
‡TA1 on MSP430x20x2 devices

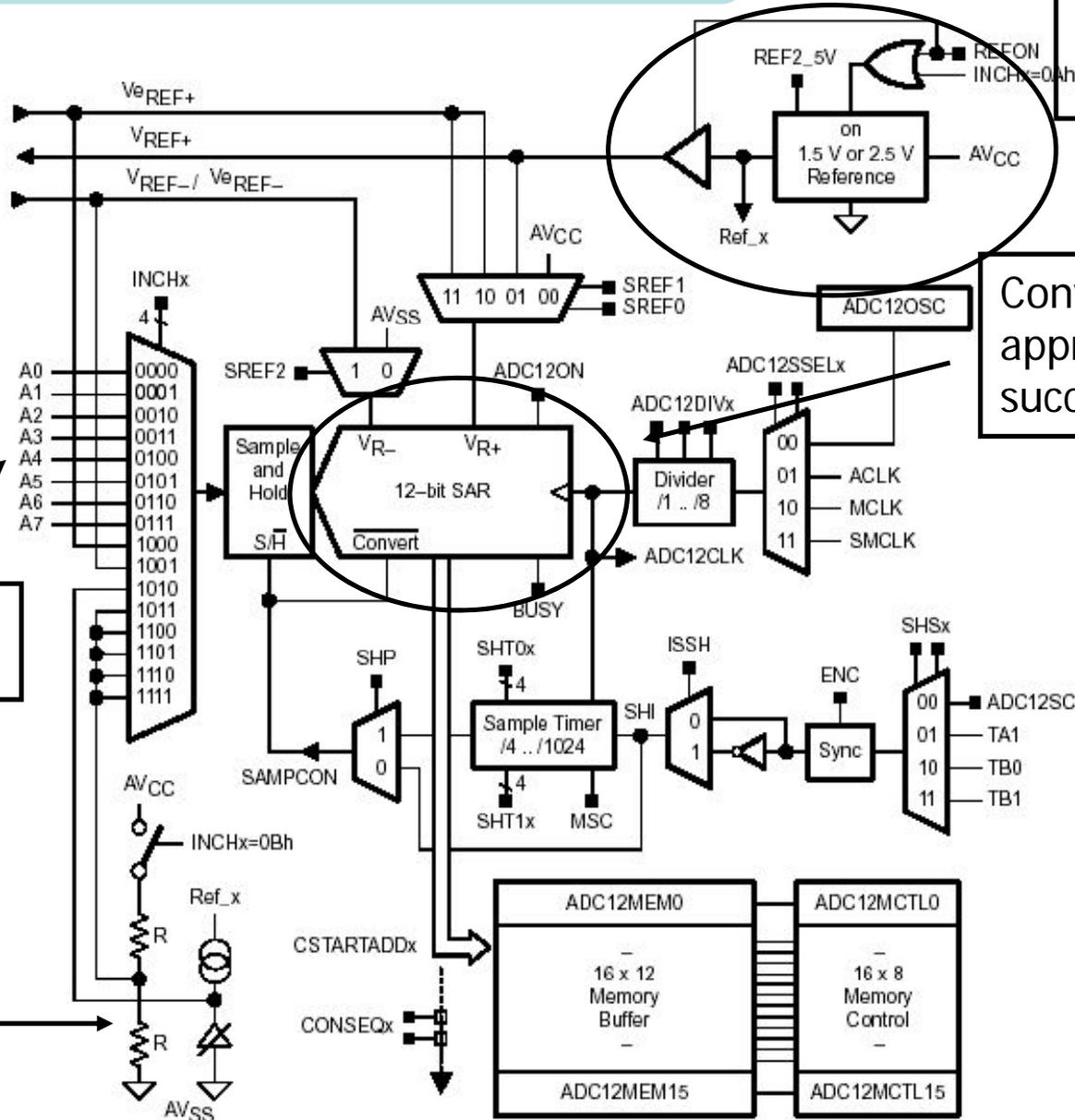
Schéma de l'ADC du MSP430FG46x

Générateur
Source de tension
référence

Convertisseur à
approximations
successives

8 entrées
analogiques

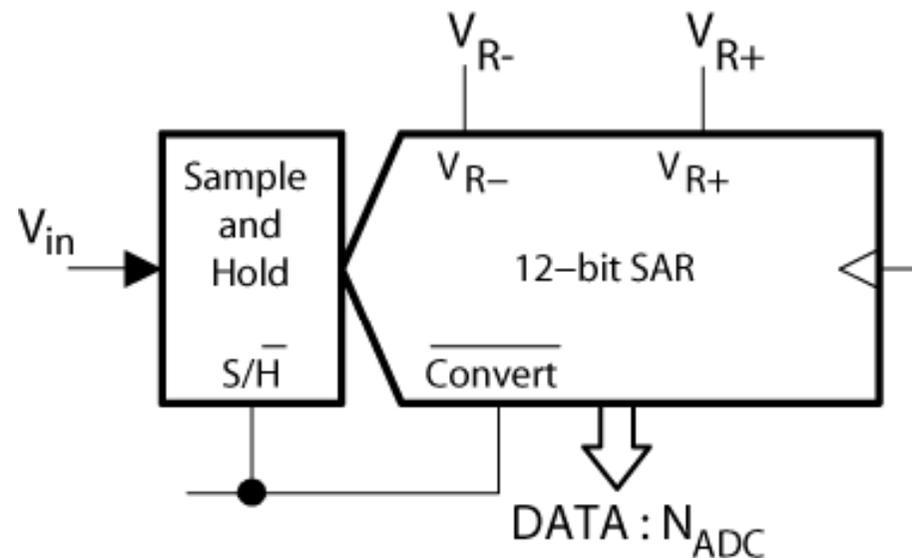
Capteur Temp°
intégré



Fonction de conversion (cas d'un ADC à 12 bits)

La fonction de conversion est donnée par la relation :

$$N_{ADC} = \left(2^{12} - 1\right) \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$



Exercices

Soit un microcontrôleur MSP430F2012.

1. Les tensions de référence du convertisseur ADC10 utilisé sur la voie A0 valent respectivement :

- $V_{R+} = 2,5 \text{ V}$
- $V_{R-} = -2,5 \text{ V}$

La tension à l'entrée du convertisseur est de 1,02 V

Quelle est la valeur numérique obtenue après conversion ?

2. Les tensions de référence du convertisseur ADC10 utilisé sur la voie A0 valent respectivement :

- $V_{R+} = 3,3 \text{ V}$
- $V_{R-} = 0 \text{ V}$

La valeur numérique obtenue est de 867.

Quelle est la tension mesurée ?

Exercices

Soit un microcontrôleur MSP430FG4617.

1. Les tensions de référence du convertisseur ADC12 utilisé sur la voie A0 valent respectivement :

- $V_{R+} = 2 \text{ V}$
- $V_{R-} = 1 \text{ V}$

La tension à l'entrée du convertisseur est de $1,2 \text{ V}$

Quelle est la valeur numérique obtenue après conversion ?

2. Les tensions de référence du convertisseur ADC12 utilisé sur la voie A0 valent respectivement :

- $V_{R+} = 3,3 \text{ V}$
- $V_{R-} = 0 \text{ V}$

La valeur numérique obtenue est de 867.

Quelle est la tension mesurée ?

Tensions de référence

Mise en veille de la référence interne

Référence supérieure interne

Référence supérieure externe

Référence inférieure externe

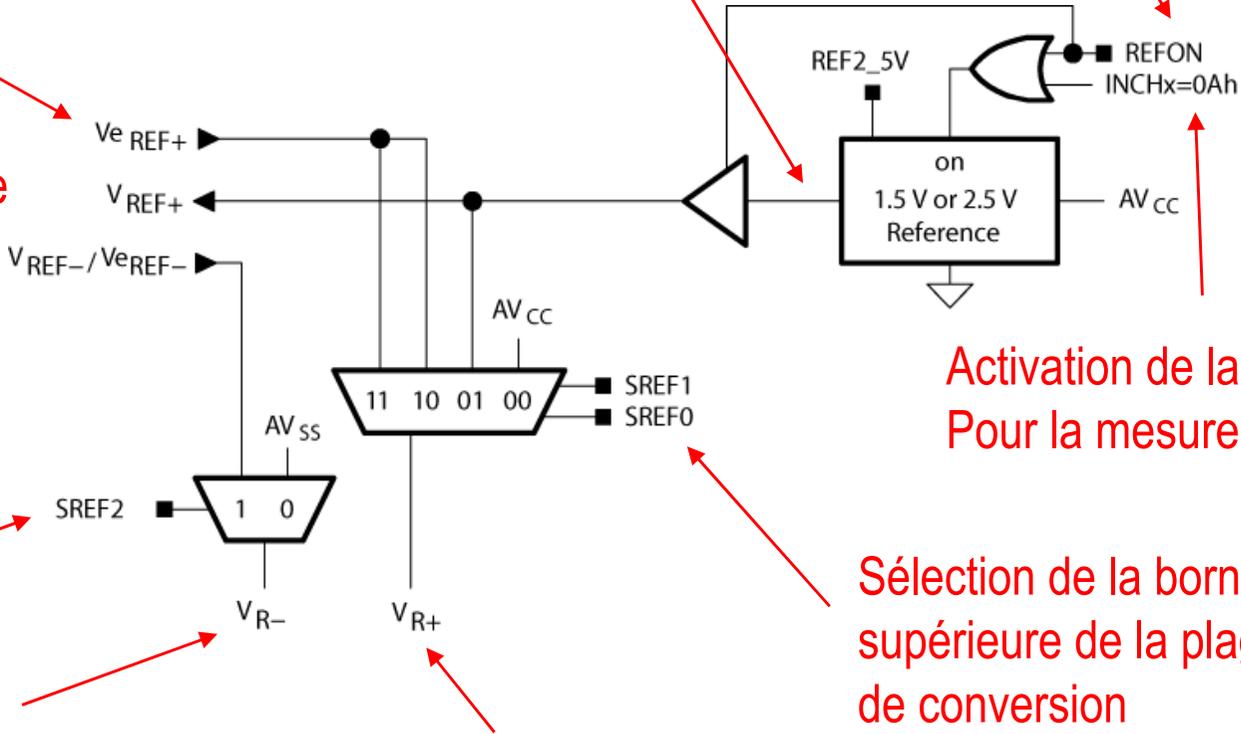
Activation de la référence
Pour la mesure de T_{int}

Sélection de la borne inférieure de la plage de conversion

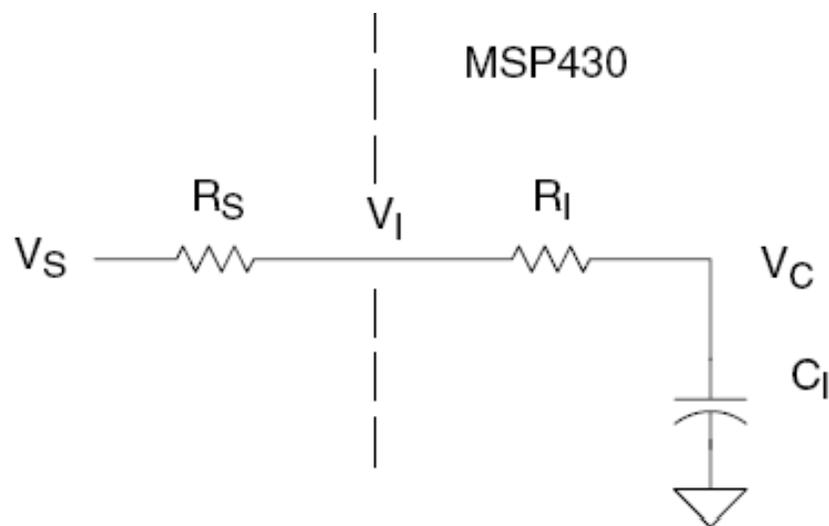
Sélection de la borne supérieure de la plage de conversion

Référence inférieure

Référence supérieure



Modèle du circuit de l'entrée analogique



V_I = Input voltage at pin Ax
 V_S = External source voltage
 R_S = External source resistance
 R_I = Internal MUX-on input resistance
 C_I = Input capacitance
 V_C = Capacitance-charging voltage

	ADC10	ADC12
R_I	2 k Ω	2 k Ω
C_I	27 pF	40 pF
t_{sampling}	$> (R_S + R_I) \times \ln(2^{11}) \times C_I$	$> (R_S + R_I) \times \ln(2^{13}) \times C_I + 800\text{ns}$

La résistance de la source influence la durée de conversion.

Exercices

1. Soit un microcontrôleur MSP430F2012.
La résistance de la source de tension analogique est de $10\text{ k}\Omega$. Calculer la durée minimale de conversion de L'ADC10.
2. Et dans le cas du MSP430FG416 (ADC12) ?

Programmation (sommaire) du convertisseur A/D

1. *Disable conversions*

1. `ADCxxCTL0 &=~ ENC;`

2. *Determine the sequence of channels*

1. `ADCxxMCTLz = SREF_x + INCH_3 + EOS;`

3. *A/D Configuration*

1. `ADCxxCTL1 = SHP + CONSEQ_1;`

2. `ADCxxCTL0 = MSH + REF2_5V + REFON + ADCxxON + ADCxxOVIE +
ADCxxTOVIE;`

4. *Wait at least ADCxx Turn On time !!!!*

5. *Enable selected interrupts*

6. *Enable conversions (ENC)*

- Les opérations de conversion dépendent d'une horloge afin de cadencer les approximations successives.
- L'horloge peut être l'une des horloges système : MCLK, ACLK, SMCLK ou un oscillateur interne ADCxxCLK .
- Le départ de conversion peut être obtenu de différentes manières:
 - le bit ADCxxSC (Start Conversion) - xx = 10, 12, ... (résolution de l'ADC)
 - la sortie du Timer A
 - la sortie du Timer (unité 0)
 - la sortie du Timer (unité 1)
- Plusieurs modes de conversion sont possibles :
 - 1 seule voie,
 - plusieurs voies séquentiellement,
 - ...

- Sources d'interruption :
 - ADCxxIFG0 - ADCxxIFG15
 - ADCxxOV, ADCxxMEMx overflow
 - ADCxxTOV, ADCxx conversion time overflow
- Les xx premiers bits sont mis à 1 quand la mémoire correspondante (de 0 à 15) est chargée par un résultat de conversion.
- Il n'y a demande d'interruption que si le bit correspondant ADCxxIE_x (et le bit GIE) sont à 1.
- Le ADCxxOV est positionné si le résultat d'une conversion intervient et mis dans une mémoire ADCxxMEM_x avant que le résultat précédent soit lu.
- Le ADCxxTOV est positionné si un ordre d'échantillonnage est donné alors que la conversion courante n'est pas terminée.

- Le registre ADCxxIFG contient les 16 flags d'interruptions (contrôlés par autant de bits du registre ADCxxIE (*Interrupt Enable*),
- qui passent à 1 quand la mémoire correspondante est remplie d'un résultat.

- Ces bits peuvent être remis à zéro par logiciel, mais sont **remis à zéro automatiquement** quand la mémoire concernée est lue.

Registres de contrôle

Jusqu'à 16 mémoires

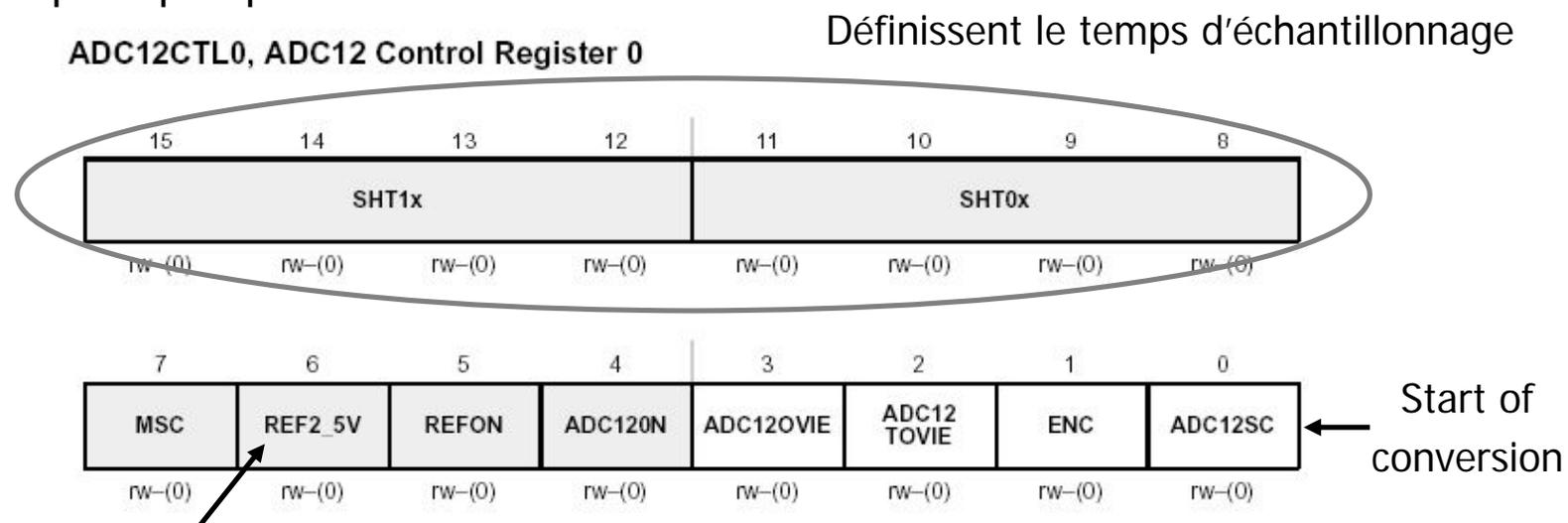
Et leur registres de contrôle associé

Register	Short Form	Register Type	Address	Initial State
ADC12 control register 0	ADC12CTL0	Read/write	01A0h	Reset with POR
ADC12 control register 1	ADC12CTL1	Read/write	01A2h	Reset with POR
ADC12 interrupt flag register	ADC12IFG	Read/write	01A4h	Reset with POR
ADC12 interrupt enable register	ADC12IE	Read/write	01A6h	Reset with POR
ADC12 interrupt vector word	ADC12IV	Read	01A8h	Reset with POR
ADC12 memory 0	ADC12MEM0	Read/write	0140h	Unchanged
ADC12 memory 1	ADC12MEM1	Read/write	0142h	Unchanged
ADC12 memory 2	ADC12MEM2	Read/write	0144h	Unchanged
ADC12 memory 3	ADC12MEM3	Read/write	0146h	Unchanged
ADC12 memory 4	ADC12MEM4	Read/write	0148h	Unchanged
ADC12 memory 5	ADC12MEM5	Read/write	014Ah	Unchanged
ADC12 memory 6	ADC12MEM6	Read/write	014Ch	Unchanged
ADC12 memory 7	ADC12MEM7	Read/write	014Eh	Unchanged
ADC12 memory 8	ADC12MEM8	Read/write	0150h	Unchanged
ADC12 memory 9	ADC12MEM9	Read/write	0152h	Unchanged
ADC12 memory 10	ADC12MEM10	Read/write	0154h	Unchanged
ADC12 memory 11	ADC12MEM11	Read/write	0156h	Unchanged
ADC12 memory 12	ADC12MEM12	Read/write	0158h	Unchanged
ADC12 memory 13	ADC12MEM13	Read/write	015Ah	Unchanged
ADC12 memory 14	ADC12MEM14	Read/write	015Ch	Unchanged
ADC12 memory 15	ADC12MEM15	Read/write	015Eh	Unchanged
ADC12 memory control 0	ADC12MCTL0	Read/write	080h	Reset with POR
ADC12 memory control 1	ADC12MCTL1	Read/write	081h	Reset with POR
ADC12 memory control 2	ADC12MCTL2	Read/write	082h	Reset with POR
ADC12 memory control 3	ADC12MCTL3	Read/write	083h	Reset with POR
ADC12 memory control 4	ADC12MCTL4	Read/write	084h	Reset with POR
ADC12 memory control 5	ADC12MCTL5	Read/write	085h	Reset with POR
ADC12 memory control 6	ADC12MCTL6	Read/write	086h	Reset with POR
ADC12 memory control 7	ADC12MCTL7	Read/write	087h	Reset with POR
ADC12 memory control 8	ADC12MCTL8	Read/write	088h	Reset with POR
ADC12 memory control 9	ADC12MCTL9	Read/write	089h	Reset with POR
ADC12 memory control 10	ADC12MCTL10	Read/write	08Ah	Reset with POR
ADC12 memory control 11	ADC12MCTL11	Read/write	08Bh	Reset with POR
ADC12 memory control 12	ADC12MCTL12	Read/write	08Ch	Reset with POR
ADC12 memory control 13	ADC12MCTL13	Read/write	08Dh	Reset with POR
ADC12 memory control 14	ADC12MCTL14	Read/write	08Eh	Reset with POR
ADC12 memory control 15	ADC12MCTL15	Read/write	08Fh	Reset with POR

Il y a quelques registres à étudier pour programmer le ADCxx:

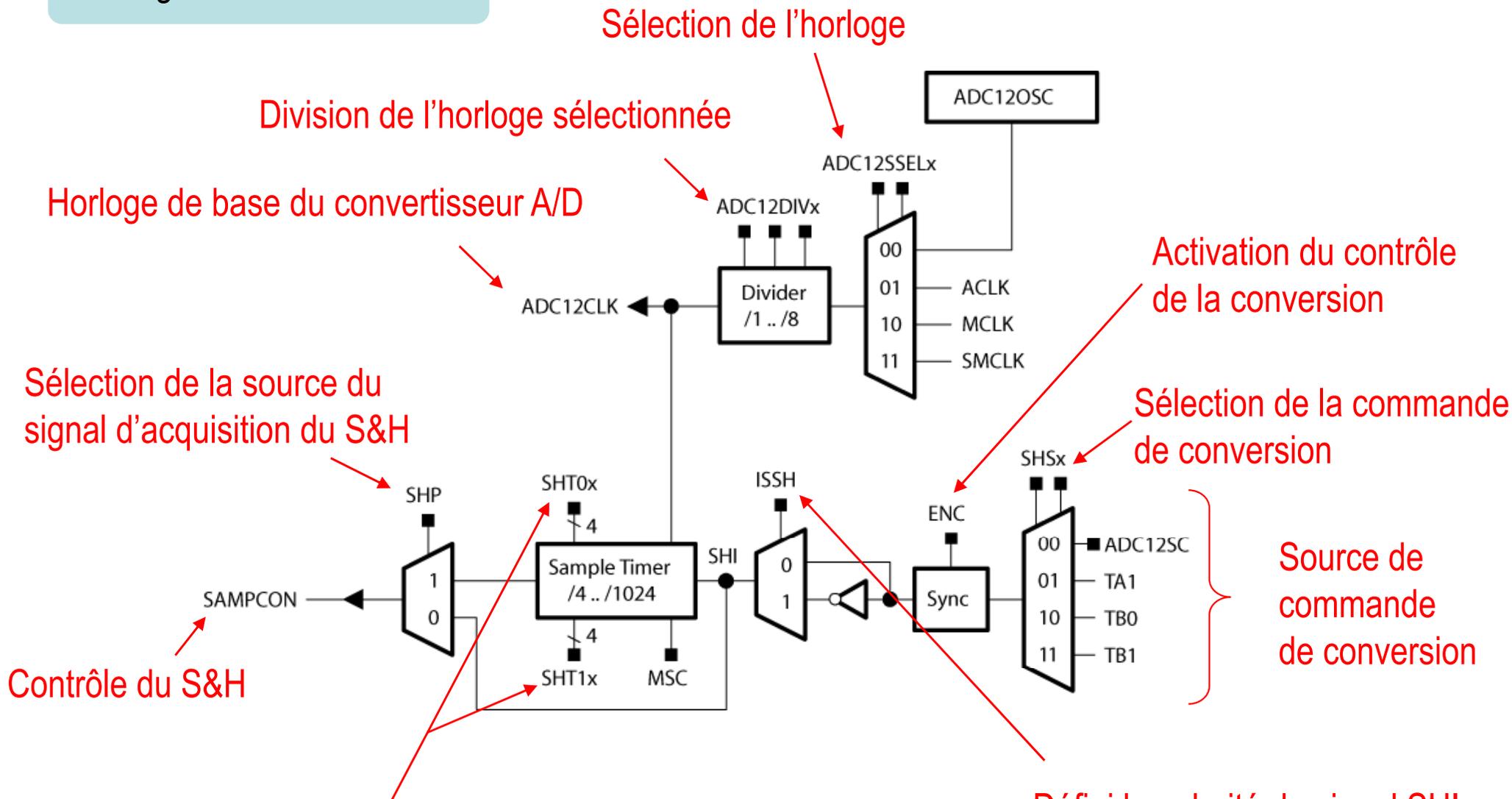
Register	Short Form	Register Type	Address	Initial State
ADC12 control register 0	ADC12CTL0	Read/write	01A0h	Reset with POR
ADC12 control register 1	ADC12CTL1	Read/write	01A2h	Reset with POR
ADC12 interrupt flag register	ADC12IFG	Read/write	01A4h	Reset with POR
ADC12 interrupt enable register	ADC12IE	Read/write	01A6h	Reset with POR
ADC12 interrupt vector word	ADC12IV	Read	01A8h	Reset with POR
ADC12 memory 0	ADC12MEM0	Read/write	0140h	Unchanged

Exemple: quelques bits de ADC12CTL0



Réf = 2.5 ou 1.5V

Timing du convertisseur A/D

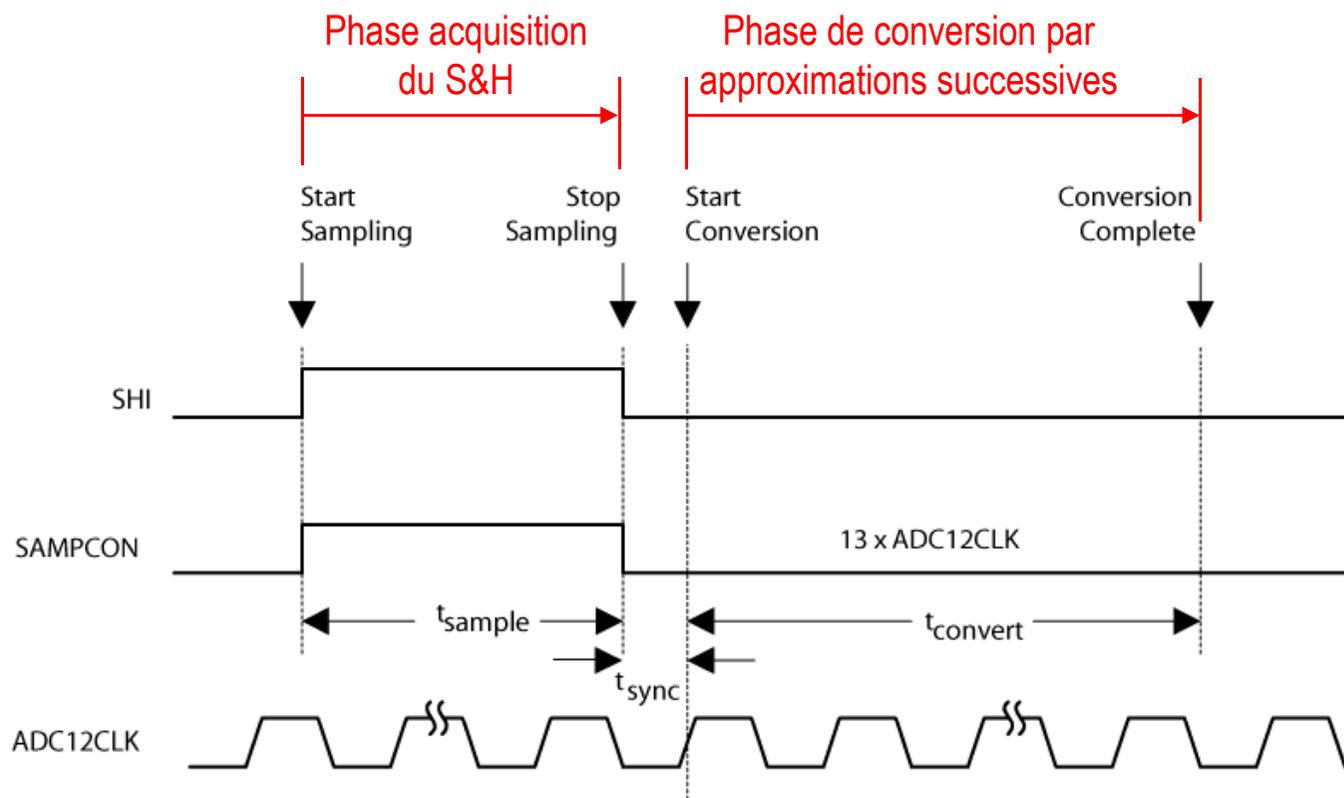


Défini le nombre de cycles d'horloge ADCxxCLK
 Pour la durée d'acquisition du S&H pour les canaux 0 - 7 et 8 - 15

Timing du convertisseur A/D

SHP=0

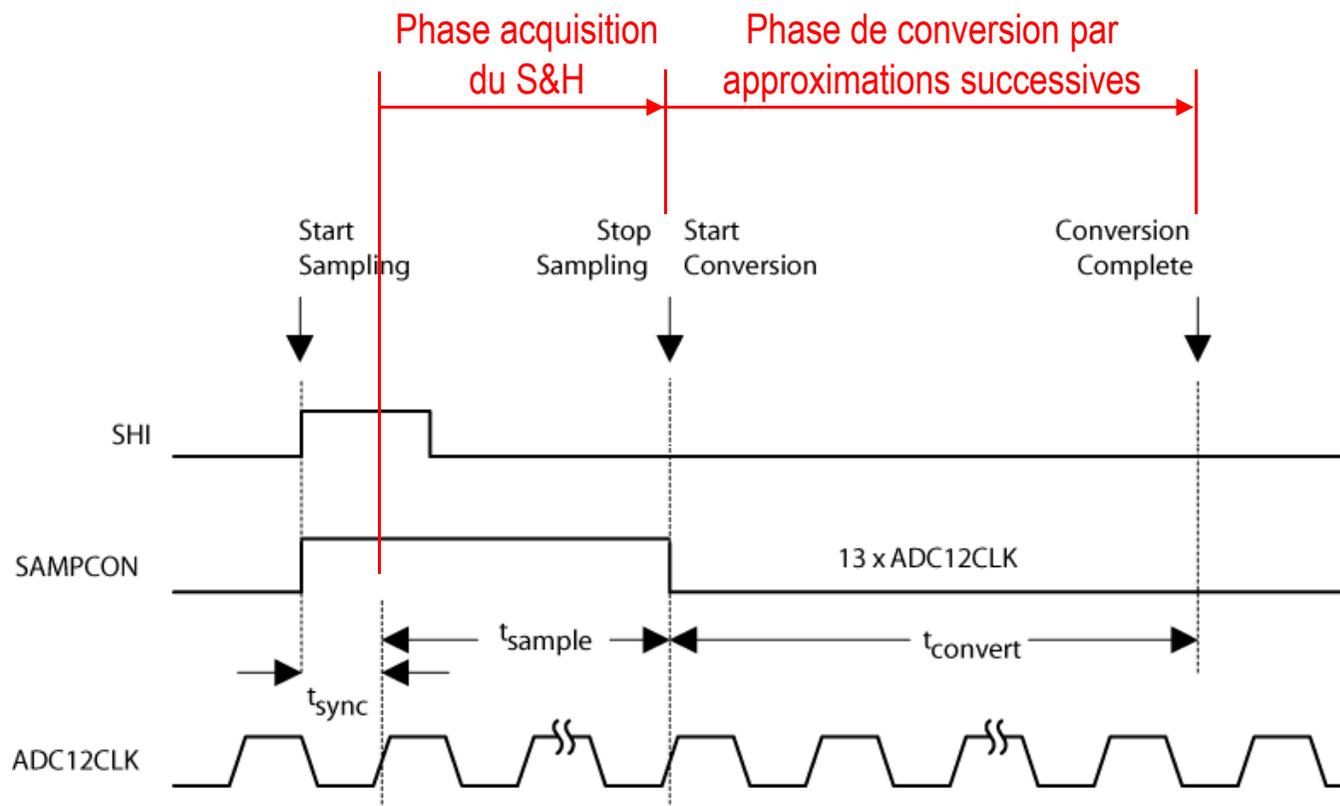
Mode d'acquisition simple. La durée d'acquisition (S&H) dépend de la source de commande, ADCxxSC, Timer_A1, (Timer_B0 ou Timer_B1, etc.)



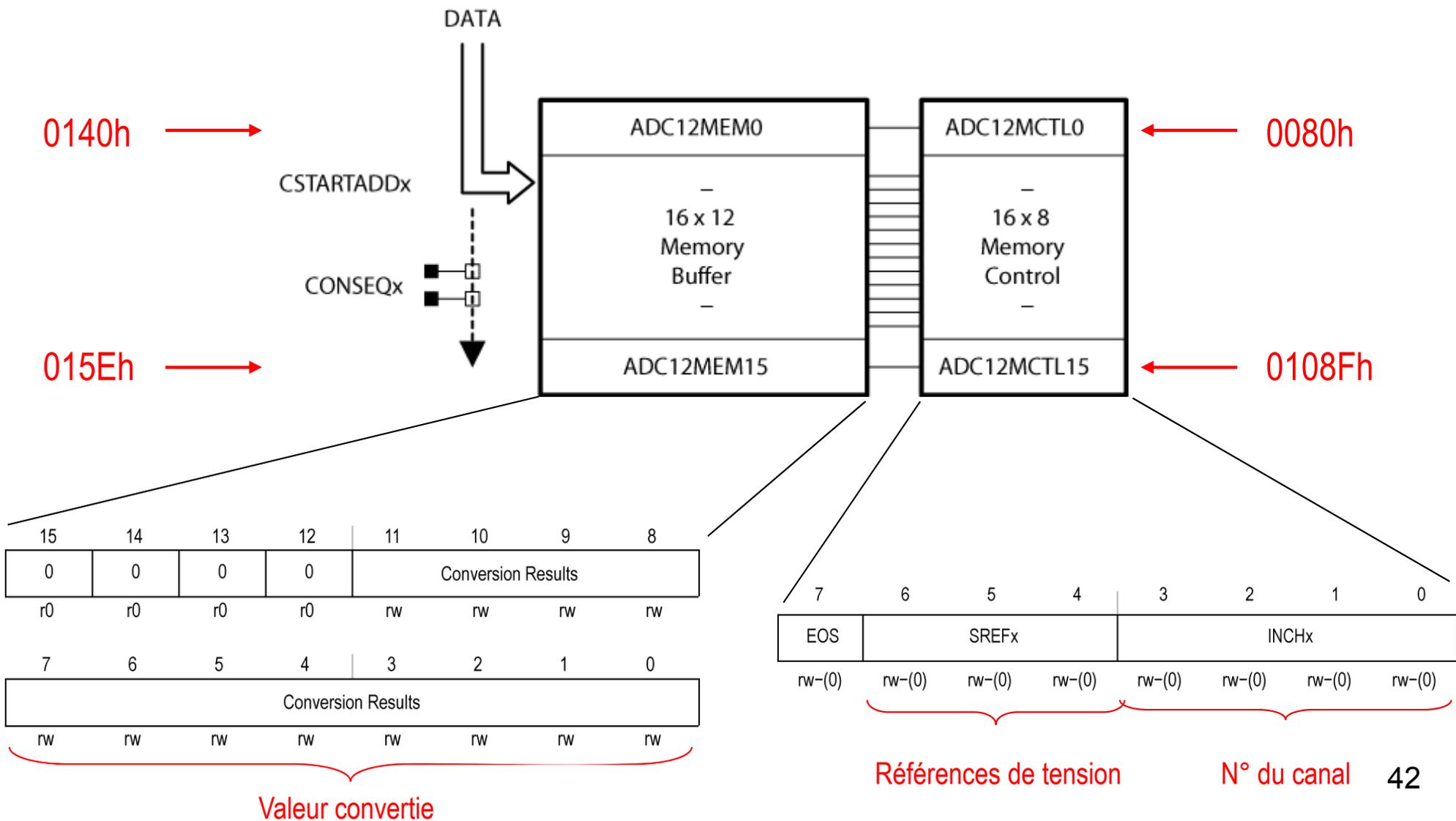
Timing du convertisseur A/D

SHP=1

Mode d'acquisition et de conversion étendu. La durée d'acquisition dépend de l'horloge ADCxxCLK et de la valeur définie dans le timer « SampleTimer ».



Buffer et gestion de la mémoire



Mode et séquence de conversion

CONSEQx	Mode	Operation
00	Un seul canal Conversion unique	Un seul canal converti une seule fois.
01	Plusieurs canaux Conversion unique	Séquence de plusieurs canaux convertis une seule fois.
10	Un seul canal Conversion répétitive	Un seul canal converti plusieurs fois.
11	Plusieurs canaux Conversion répétitive	Séquence de plusieurs canaux convertis plusieurs fois.

Bits 1 et 2 du registre ADCxxCTL1

Conversion simple d'une seule voie

CONSEQx = 00

Contient la position x du buffer ADCxxMEM dans laquelle sera placé le résultat de la conversion

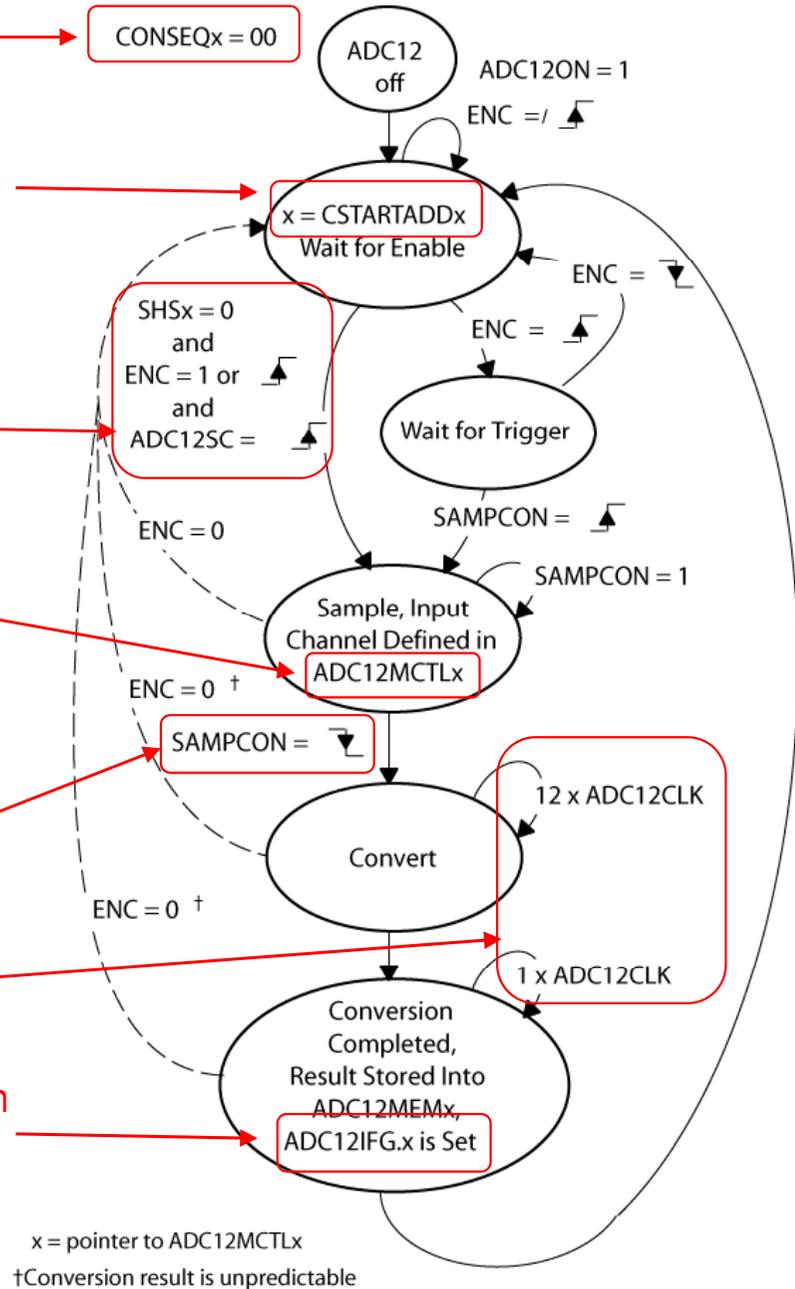
Exemple d'un déclenchement software (ADCxxSC) d'une conversion

Le canal à convertir se trouve dans le registre ADCxxMCTLx (INCHx)

Fin de l'acquisition du S&H, démarrage de la conversion par approximations successives

Résultat accessible après 13 cycles d'horloge ADCxxCLK

Indique qu'une nouvelle conversion est à disposition dans le buffer ADCxxMEM en position x



x = pointer to ADC12MCTLx
†Conversion result is unpredictable

Conversion séquentielle répétitive de toutes les voies

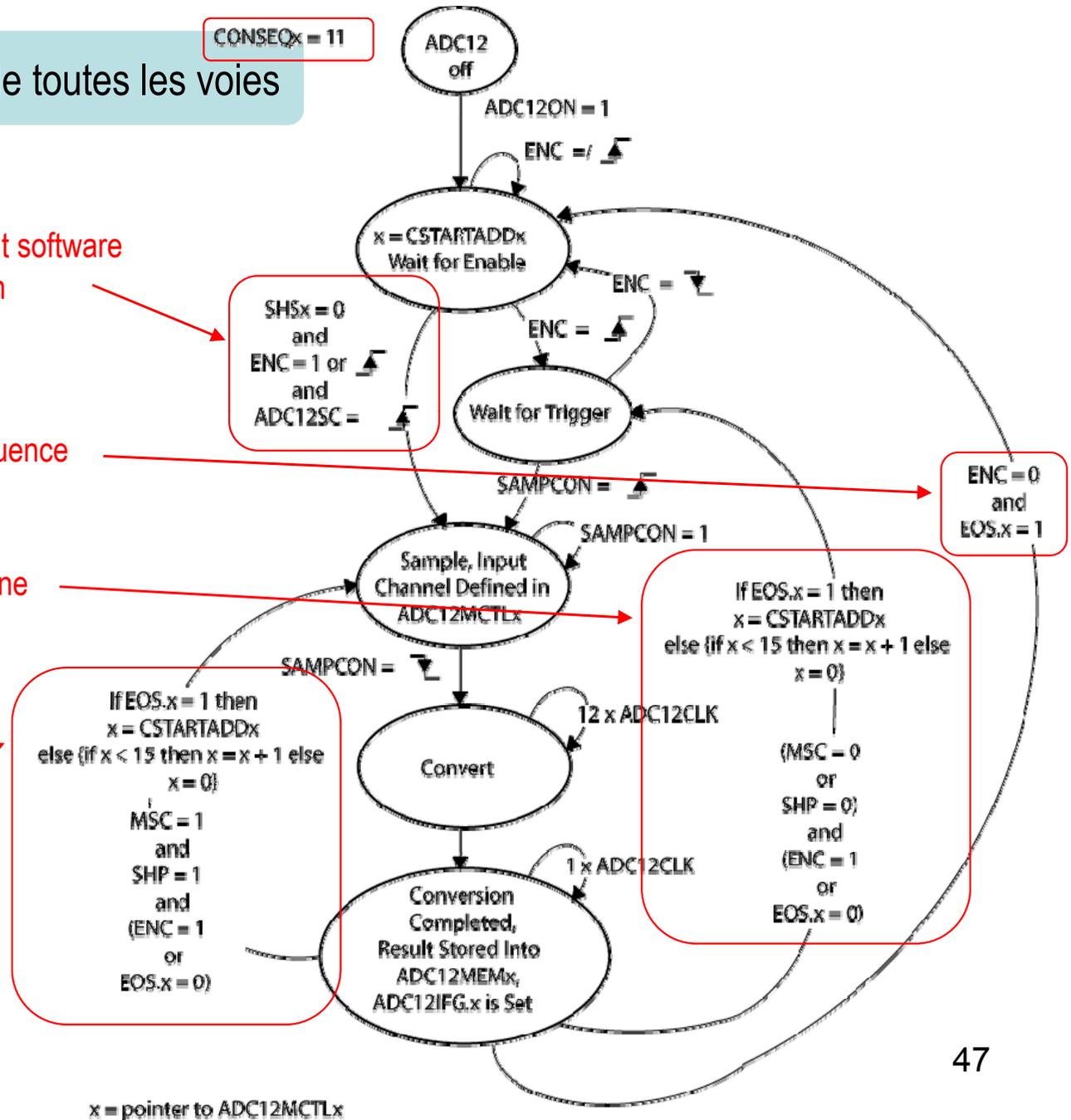
CONSEQx = 11

Exemple d'un déclenchement software (ADCxxSC) d'une conversion

Indique la fin de la séquence

L'acquisition du canal suivant ne se fera qu'après un flanc montant de SHI, à la fin d'une séquence (EOS.x=1) il y a répétition de la séquence

L'acquisition du canal suivant est automatiquement initié, à la fin d'une séquence (EOS.x=1) il y a répétition de la séquence



Sources d'interruption

Le périphérique ADCxx à jusqu'à 18 sources d'interruption

ADCxxIFG0 – ADCxxIFG15 :

Indique qu'une nouvelle valeur convertie est chargée dans ADCxxMEMx .

Interruption active que si GIE (General Interrupt Enable) est 1.

ADCxxOV : Indique qu'une nouvelle valeur convertie est chargée dans ADCxxMEMx
Alors que la valeur précédente n'a pas été lue.

ADCxxOT : Indique qu'une nouvelle conversion est demandée alors que la précédente est en cours.

Emplacement du vecteur d'interruption ADCxx dans la table des vecteurs d'interruption

Toutes les sources d'interruption ADCxx sont regroupées en un seul vecteur d'interruption

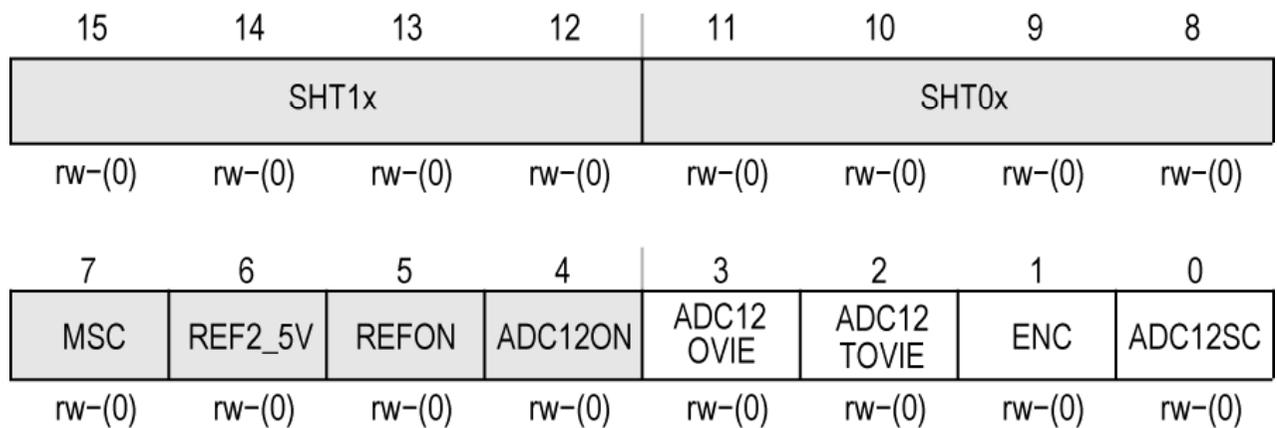
INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Flash Memory	WDTIFG KEYV	Reset	0FFFEh	31, highest
NMI Oscillator Fault Flash Memory Access Violation	NMIIFG OFIFG ACCVIFG	(Non)maskable (Non)maskable (Non)maskable	0FFFCh	30
Timer_B7	TBCCR0 CCIFG0	Maskable	0FFFAh	29
Timer_B7	TBCCR1 CCIFG1 ... TBCCR6 CCIFG6, TBIFG	Maskable	0FFF8h	28
Comparator_A	CAIFG	Maskable	0FFF6h	27
Watchdog Timer+	WDTIFG	Maskable	0FFF4h	26
USCI_A0/USCI_B0 Receive	UCA0RXIFG, UCB0RXIFG	Maskable	0FFF2h	25
USCI_A0/USCI_B0 Transmit	UCA0TXIFG, UCB0TXIFG	Maskable	0FFF0h	24
ADC12	ADC12IFG	Maskable	0FFEEh	23
Timer_A3	TACCR0 CCIFG0	Maskable	0FFECh	22
Timer_A3	TACCR1 CCIFG1 and TACCR2 CCIFG2, TAIFG	Maskable	0FFEAh	21
I/O Port P1 (Eight Flags)	P1IFG.0 to P1IFG.7	Maskable	0FFE8h	20
USART1 Receive	URXIFG1	Maskable	0FFE6h	19
USART1 Transmit	UTXIFG1	Maskable	0FFE4h	18
I/O Port P2 (Eight Flags)	P2IFG.0 to P2IFG.7	Maskable	0FFE2h	17
Basic Timer1/RTC	BTIFG	Maskable	0FFE0h	16
DMA	DMA0IFG, DMA1IFG, DMA2IFG	Maskable	0FFDEh	15
DAC12	DAC12.0IFG, DAC12.1IFG	Maskable	0FFDCh	14
Reserved..	Reserved		0FFDAh	13
		
			0FFC0h	0, lowest

Registre des sources d'interruption : ADCxxCTL0

Le registre ADCxxIV contient les indications concernant la source d'interruption à l'intérieur du périphérique ADCxx

ADC12IV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
000h	No interrupt pending		
002h	ADC12MEMx overflow		Highest
004h	Conversion time overflow		
006h	ADC12MEM0 interrupt flag	ADC12IFG0	
008h	ADC12MEM1 interrupt flag	ADC12IFG1	
00Ah	ADC12MEM2 interrupt flag	ADC12IFG2	
00Ch	ADC12MEM3 interrupt flag	ADC12IFG3	
00Eh	ADC12MEM4 interrupt flag	ADC12IFG4	
010h	ADC12MEM5 interrupt flag	ADC12IFG5	
012h	ADC12MEM6 interrupt flag	ADC12IFG6	
014h	ADC12MEM7 interrupt flag	ADC12IFG7	
016h	ADC12MEM8 interrupt flag	ADC12IFG8	
018h	ADC12MEM9 interrupt flag	ADC12IFG9	
01Ah	ADC12MEM10 interrupt flag	ADC12IFG10	
01Ch	ADC12MEM11 interrupt flag	ADC12IFG11	
01Eh	ADC12MEM12 interrupt flag	ADC12IFG12	
020h	ADC12MEM13 interrupt flag	ADC12IFG13	
022h	ADC12MEM14 interrupt flag	ADC12IFG14	
024h	ADC12MEM15 interrupt flag	ADC12IFG15	Lowest

Registre de contrôle: ADC12CTL0 (MSP430FG416)



Modifiable seulement lorsque ENC = 0

SHT1x : nombre de cycles d'horloge ADCxxCLK pour la durée de la phase d'acquisition du S&H pour les canaux 15 à 8

SHT0x : nombre de cycles d'horloge ADCxxCLK pour la durée de la phase d'acquisition du S&H pour les canaux 7 à 0

SHTx Bits	ADC12CLK cycles
0000	4
0001	8
0010	16
0011	32
0100	64
0101	96
0110	128
0111	192
1000	256
1001	384
1010	512
1011	768
1100	1024
1101	1024
1110	1024
1111	1024

Registre de contrôle : ADC12CTL0

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)



Modifiable lorsque ENC = 0

- MCS :** Conversion multiple (mode séquentielle ou / et répétitif)
 0 : Le « Sample Timer » est déclenché par un flanc montant de SHI
 1 : Le premier flanc montant de SHI déclenche le « Sample Timer » les autres conversion se font automatiquement
- REF2_5V :** Choix de la référence de tension (REFON doit être mis à 1)
 0 : Référence 1.5V
 1 : Référence 2.5V
- REFON :** Référence de tension ON (alimentation de la référence de tension interne)
 0 : Référence OFF (désactivé)
 1 : Référence ON (activé)

Registre de contrôle : ADC12CTL0

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)



Modifiable lorsque ENC = 0

ADC12ON : ADCxx ON (alimentation du convertisseur AD)
 0 : ADCxx OFF (désactivé)
 1 : ADCxx ON (activé)

ADC12OVIE : Autorisation d'une interruption due a une écriture dans le buffer de résultat
 (ADCxxMEMx) avant la lecture de la valeur précédente (le flag GIE doit être à 1).
 0 : interruption non autorisée
 1 : interruption autorisée

Registre de contrôle : ADC12CTL0

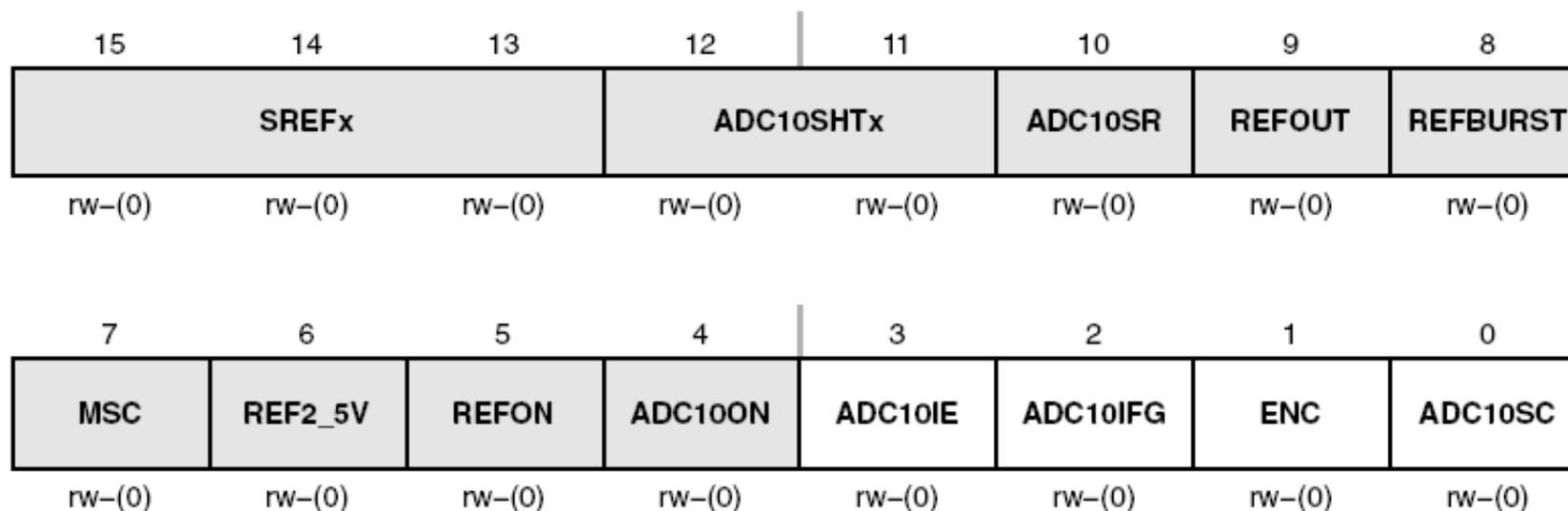
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)



Modifiable lorsque ENC = 0

- ADC12TOVIE :** Autorisation d'une interruption due a une demande de conversion avant la fin de la conversion précédente (le flag GIE doit être à 1).
 0 : interruption non autorisée
 1 : interruption autorisée
- ENC :** Autorisation d'une commande de démarrage de conversion
 0 : commande de conversion pas autorisée
 1 : commande de conversion autorisée
- ADC12SC :** Contrôle software du démarrage d'un conversion
 Peut être activé simultanément avec ENC
 0 : pas de conversion demandée
 1 : demande d'une conversion (automatiquement remis à 0)

Registre de contrôle: ADC10CTL0 (MSP430F2012)



Modifiable only when ENC = 0

Les définitions sont semblables, mais pas identiques.
Voir le manuel de programmation du F201x, chapitre 20.

Registre de contrôle : ADC12CTL1 (MSP430FG416)

15	14	13	12	11	10	9	8
CSTARTADDx				SHSx		SHP	ISSH
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12DIVx			ADC12SSELx		CONSEQx		ADC12 BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)



Modifiable seulement lorsque ENC = 0

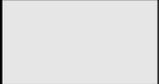
CSTARTADDx : Voie (*channel*) de départ de la conversion

Conversion simple : buffer utilisé pour sauver le résultat

Conversion séquentielle ou répétitive : buffer de mémorisation du premier résultat

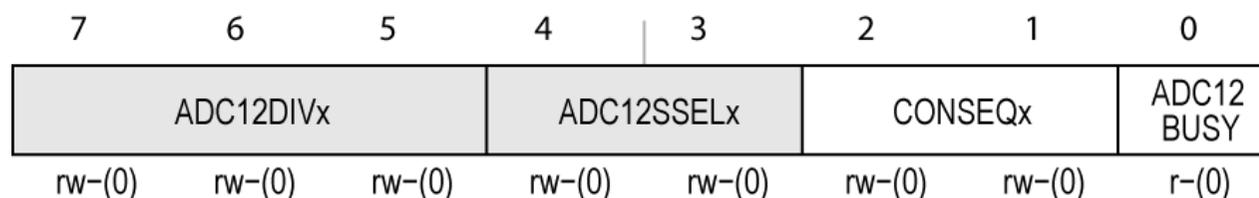
Registre de contrôle : ADC12CTL1



 Modifiable seulement lorsque ENC = 0

- SHSx :** Source du contrôle du S&H et du démarrage de la conversion
 00 : ADC12SC (commande software, registre de contrôle ADCxxTL0[0])
 01 : Timer_A.OUT1
 10 : Timer_A.OUT0
 11 : Timer_B.OUT1
- SHP :** Mode d'acquisition simple ou étendu
 0 : signal SAMPCON identique à SHI (mode simple)
 1 : signal SAMPCON étendu par « SAMPLE TIMER
- ISSH :** Inversion de la polarité du signal SHI
 0 : signal non inversé
 1 : signal inversé

Registre de contrôle : ADCxxCTL1



Modifiable seulement lorsque ENC = 0

ADC12DIVx : Division de l'horloge de l'ADCxx (ADCxxOSC, ACLK, MCLK, SMCLK)

000 : /1
 001 : /2
 010 : /3
 011 : /4
 100 : /5
 101 : /6
 110 : /7
 111 : /8

ADC12BUSY : Convertisseur occupé soit par une acquisition sur le S&H, soit par une conversion

0 : convertisseur prêt
 1 : convertisseur occupé

Registre de contrôle : ADCxxCTL1



Modifiable seulement lorsque ENC = 0

ADC12SSELx : Source de l'horloge du périphérique ADCxx

00 : ADC12OSC

01 : ACLK

10 : MCLK

11 : SMCLK

CONSEQx : Sélection du mode de conversion

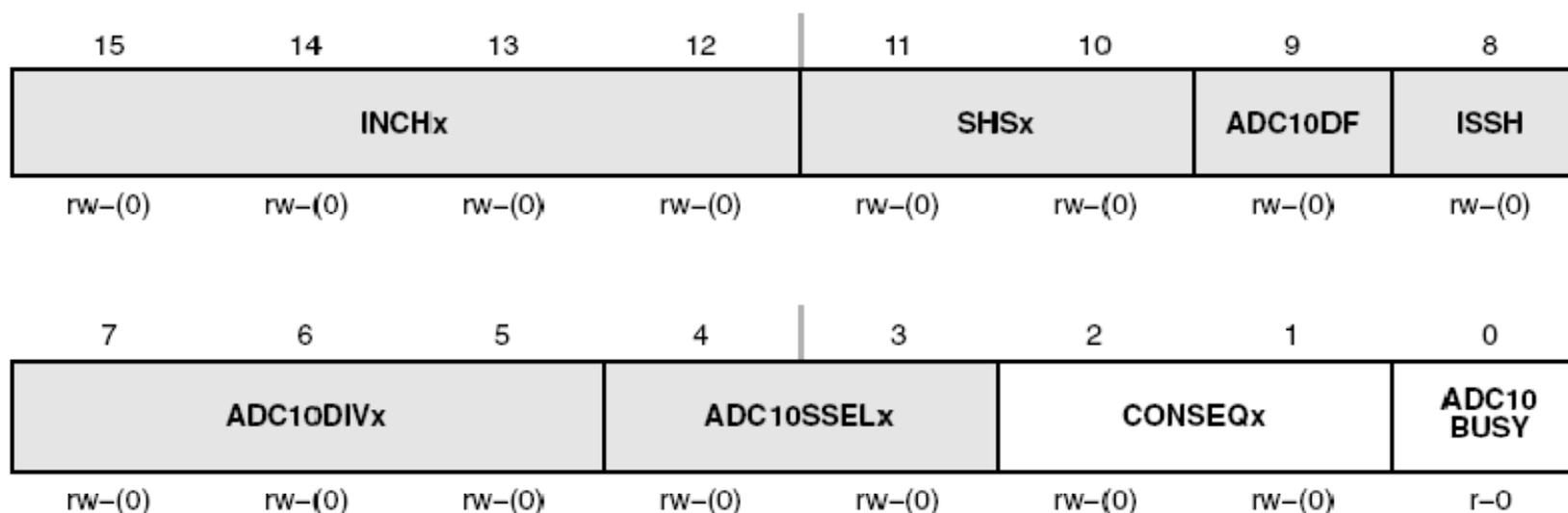
00 : conversion simple d'un seul canal

01 : conversion séquentielle de plusieurs canaux

10 : conversion répétitive d'un seul canal

11 : conversion séquentielle répétitive de plusieurs canaux

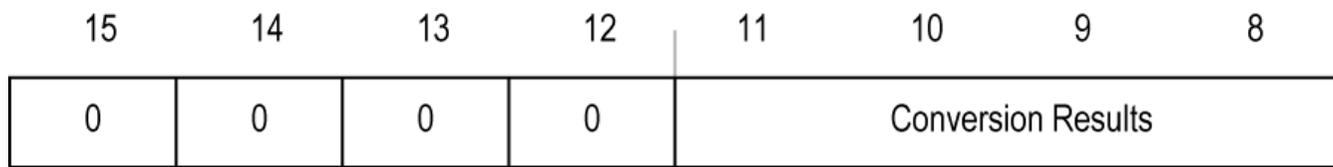
Registre de contrôle : ADC10CTL1 (MSP430F201x)



Modifiable only when ENC = 0

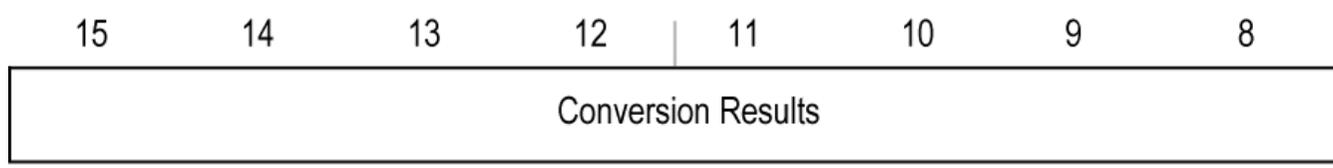
Les définitions sont semblables, mais pas identiques.
Voir le manuel de programmation du F201x, chapitre 20.

Registres du buffer : ADCxxMEMx

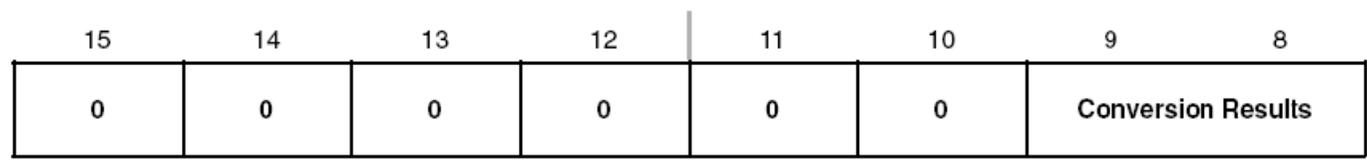


r0 r0 r0 r0 rw rw rw rw

ADC12

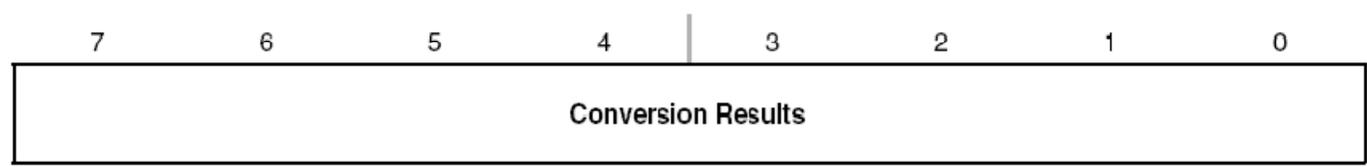


rw rw rw rw rw rw rw rw



r0 r0 r0 r0 r0 r0 r r

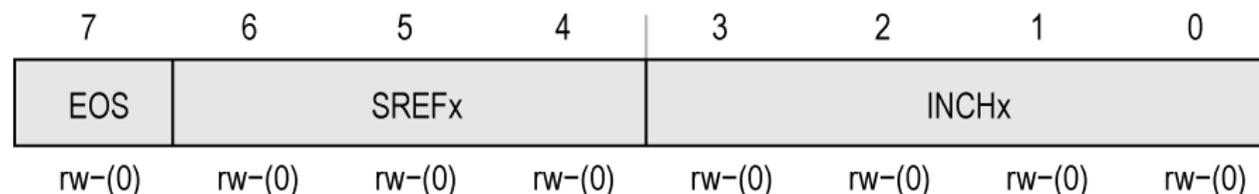
ADC10



r r r r r r r r

Le résultat est sur xx bit. Ces derniers sont justifiés à droite, les quatre (six) bits de poids fort sont à 0

Registre de contrôle du buffer du résultat X: ADC12MCTL_X



 Modifiable seulement lorsque ENC = 0

EOS : Fin de séquence. Indique la dernière conversion d'une séquence

0 : pas de fin de séquence

1 : fin de séquence

SREFx : Choix de la référence

000 : $V_{R+} = AV_{CC}$ et $V_{R-} = AV_{SS}$

001 : $V_{R+} = V_{REF+}$ et $V_{R-} = AV_{SS}$

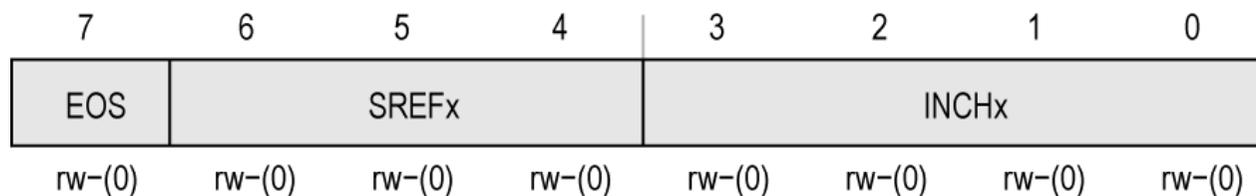
01 Φ : $V_{R+} = V_{eREF+}$ et $V_{R-} = AV_{SS}$

100 : $V_{R+} = AV_{CC}$ et $V_{R-} = V_{REF-} / V_{eREF-}$

101 : $V_{R+} = V_{REF+}$ et $V_{R-} = V_{REF-} / V_{eREF-}$

11 Φ : $V_{R+} = V_{eREF+}$ et $V_{R-} = V_{REF-} / V_{eREF-}$

Registre de contrôle : ADC12MCTL



 Modifiable seulement lorsque ENC = 0

INCH :

Sélection d'un canal

0000 : A0

0001 : A1

0010 : A2

0011 : A3

0100 : A4

0101 : A5

0110 : A6

0111 : A7

1000 : V_{REF+} 1001 : V_{REF-} / V_{eREF-}

1010 : Température

1011 : $(AV_{CC} - AV_{SS}) / 2$

1100 : A12

1101 : A13

1110 : A14

1111 : A15

Registre de contrôle d'autorisation d'interruption : ADC12IE

15	14	13	12	11	10	9	8
ADC12IE15	ADC12IE14	ADC12IE13	ADC12IE12	ADC12IE11	ADC12IE10	ADC12IE9	ADC12IE8
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	7	7	7	7	7	7	7
ADC12IE7	ADC12IE6	ADC12IE5	ADC12IE4	ADC12IE3	ADC12IE2	ADC12IE1	ADC12IE0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

ADC12IE_x : Interruption x autorisée.
 0 : interruption non autorisée
 1 : interruption autorisée

Registre de contrôle d'autorisation d'interruption : ADC12IFG

15	14	13	12	11	10	9	8
ADC12 IFG15	ADC12 IFG14	ADC12 IFG13	ADC12 IFG12	ADC12 IFG11	ADC12 IFG10	ADC12 IFG9	ADC12 IFG8
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12 IFG7	ADC12 IFG6	ADC12 IFG5	ADC12 IFG4	ADC12 IFG3	ADC12 IFG2	ADC12 IFG1	ADC12 IFG0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

ADC12IFGx : Flag mis à 1 lorsqu'une valeur est chargée dans le registre correspondant ADC12MEMx du buffer ADC12MEM.

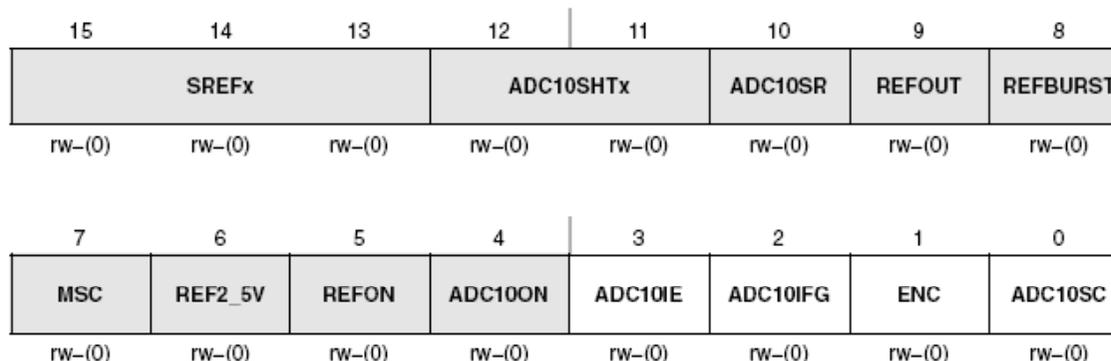
Le flag est remis à 0 lorsque la valeur est lue:

0 : pas d'interruption pendante

1 : interruption pendante

Résumé: registres de l'ADC10 (MSP430F2012)

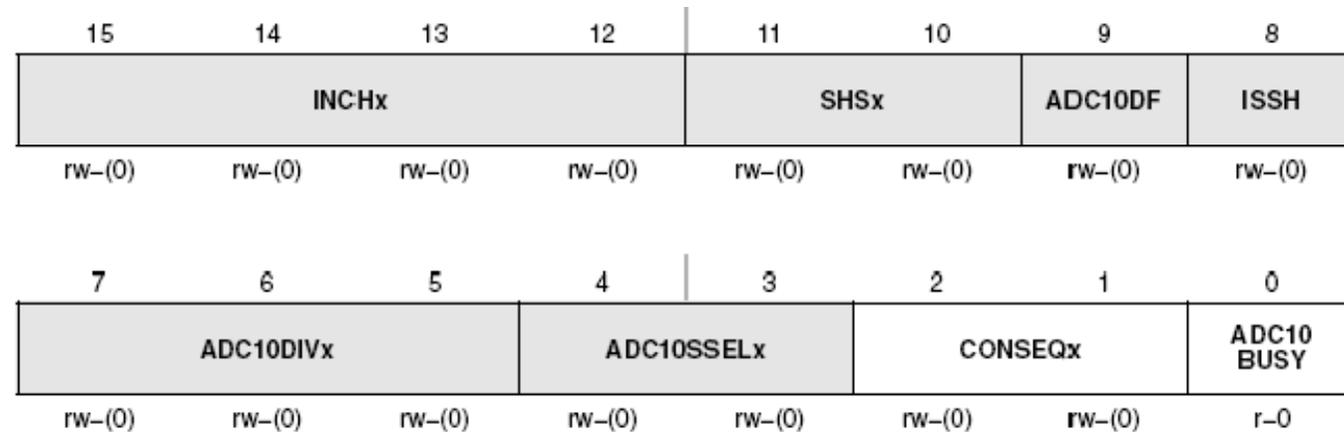
ADC10CTL0



SREF_X	→	Choix des références de tension V_{R+} et V_{R-}
ADC10SHT	→	Durée d'acquisition
ADC10SR	→	Fréquence d'échantillonnage
REFOUT	→	Sortie référence on/off
REFBURST	→	«Reference burst»
MSC	→	Conversion multiple
REF2_5V	→	Référence 2,5V (sinon 1,5V)
REFON	→	Référence active (ON)
ADC10ON	→	Ecriture overflow enable
ADC10IE	→	Interruption activée
ADC10IFG	→	Flag d'interruption
ENC	→	Autorise conversion
ADC10SC	→	Démarre conversion

Résumé: registres de l'ADC10

ADC10CTL1



INCH	→	Voie (<i>channel</i>) convertie
SHS_X	→	Source du démarrage de la conversion
ADC10DF	→	Format (binaire simple ou complément à 2)
ISSH	→	Signal inversé
ADC10DIV_X	→	Diviseur de l'horloge
ADC10SSEL_X	→	Source de l'horloge
CONSEQ_X	→	Mode de conversion
ADC10BUSY	→	ADC10 occupé

Résumé: registres de l'ADC10

ADC10AEx	→	« Analog enable »
ADC10MEM	→	Buffer du résultat
ADC10DTCx	→	Transfer direct
ADC10SA	→	Adresse initiale pour DTC

Exemple en C

```
include "io430.h"

int main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTXLD;

    ADC10CTL0 = ADC10SHT_2 + ADC10ON + REF2_5V + REFON;
    ADC10CTL1 = INCH 1;

    P1DIR |= 0x01; // P1.0 = output
    P1SEL |= 0x02;
    // ou équivalent ADC10AE0 |= 0x02; // P1.1 ADC option select
    P1OUT = 0;

    while(1)
    {
        ADC10CTL0 |= ENC + ADC10SC; // Lancer échantillonnage
        while (ADC10CTL1 & ADC10BUSY); // on attend un peu ...
        if (ADC10MEM > 512) P1OUT = 0x01 ;
        else P1OUT = 0 ;
    }
}
```

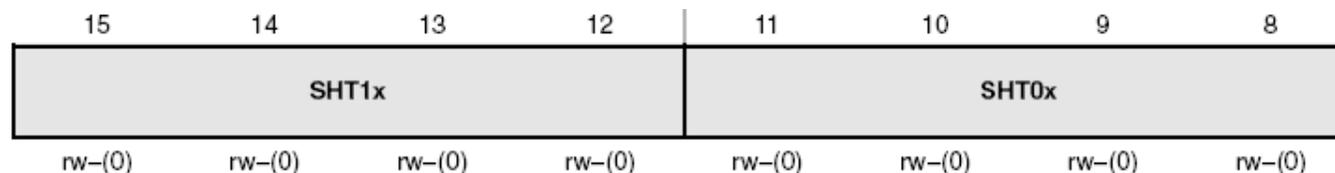
Résumé: registres de l'ADC12 du MSP430FG4617

ADC12CTL0

SHT0_x, SHT1_x



Durée d'acquisition



MSC



Conversion multiple

REF2_5V



Référence 2,5V (sinon 1,5V)

REFON



Référence active (ON)

ADC12ON



Ecriture overflow enable

ADC12OVIE, ADC12TOVIE



Conversion overflow enable

ENC



Autorise conversion

ADC12SC



Démarre conversion



- SHT1x** Bits 15-12 Sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15.
- SHT0x** Bits 11-8 Sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7.

SHTx Bits	ADC12CLK cycles
0000	4
0001	8
0010	16
0011	32
0100	64
0101	96
0110	128
0111	192
1000	256
1001	384
1010	512
1011	768
1100	1024
1101	1024
1110	1024
1111	1024

Registres de l'ADC12 du MSP430FG4617

ADC12CTL1

CSTARTADD_X	→	Signal (<i>channel</i>) de départ (entre 0 et 15)
SHS_X	→	Source du démarrage de la conversion
SHP	→	Mode d'acquisition étendu
ISSH	→	Signal inversé
ADC12DIV_X	→	Diviseur de l'horloge
ADC12SSEL_X	→	Source de l'horloge
CONSEQ_X	→	Mode de conversion

ADC12MCTL_X	→	Registre de contrôle du buffer du résultat X
EOS	→	Fin de séquence
SREF_X	→	Choix des références de tension V_{R+} et V_{R-}
INCH	→	Voie (<i>channel</i>) convertie

$$N_{ADC} = (2^{12} - 1) \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

ADC12MEM_X	→	Buffer du résultat
ADC12IE	→	Interruption activée
ADC12IFG	→	Flag d'interruption

Exemple en C

Le MSP430FG461x comprend un capteur analogique de température interne, qui peut être lu sur la voie 10.

```

//*****
//  MSP430xG461x - ADC12, Sample A10 Temperature
//  Set P2.x (LEDs) if Temp ++ ~2C
//
//  Description: Use ADC12 and the integrated temperature sensor to detect
//  temperature gradients. The temperature sensor output voltage is sampled
//  ~ every 80ms and compared with the defined delta values using an ISR.
//  (ADC12OSC/256)/ determines sample time which needs to be greater than
//  30us for temperature sensor.
//
//  ADC12 is operated in repeat-single channel mode with the sample and
//  convert trigger sourced from Timer_A CCR1. The ADC12MEM0_IFG at the end
//  of each conversion will trigger an ISR.
//
//  ACLK = 32kHz, MCLK = SMCLK = default DCO 1048576Hz, ADC12CLK = ADC12OSC
//
//
//          MSP430xG461x
//          -----
//          /|\|          XIN|-
//          | |          | 32kHz
//          --|RST      XOUT|-
//          |          |
//          |          P2.x|-->LED
//
//*****

```

Exemple en C

```

#include "msp430xG46x.h"
#include "intrinsics.h"

#define ADC_DELTA_ON 12 // ~ 2 Deg C delta for LED on
unsigned int ADCResult;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog

    ADC12CTL0 = SHT0_8 + REF2_5V + REFON + ADC12ON; // Config ADC12
    ADC12CTL1 = SHS_1 + SHP + CONSEQ_2; // TA trig.,
    // Sample/Hold Pulse Mode
    // Repetitive conversion of 1 channel

    ADC12MCTL0 = SREF_1 + INCH_10; // Channel A10, Vref+
    ADC12IE |= 0x001; // Enable ADC12IFG.0

    // fréquence du timer = 32*32768/CCR0 [Hz]
    TACCR0 = 13600; // Delay for reference start-up
    TACTL = TACLRL + MC_1 + TASSEL_2; // up mode, SMCLK
    TACTL = TASSEL_2 + MC_2; // SMCLK, cont-mode
    TACCTL1 = OUTMOD_4; // Toggle on EQU1 (TAR = 0)

    ADC12CTL0 |= ENC; // Enable ADC12

    while (!(ADC12IFG & 0x0001)); // Wait for first conversion?

    ADCResult = ADC12MEM0; // Read out 1st ADC value
    ADCResult += ADC_DELTA_ON;
    P2OUT = 0; // Clear P5
    P2DIR |= 0xCF; // P5.1 as output

    // ... suite ...

```

Exemple en C

... suite et fin ...

```
__enable_interrupt();
while (1);
}

#pragma vector = ADC12_VECTOR
__interrupt void ADC12_ISR(void)
{
    if (ADC12MEM0 >= ADCResult)           // ADC12MEM = A0 > ADCResult ?
        P2OUT |= 0xCF;                   // P2.x = 1
    else
        P2OUT &= ~0xCF;                   // P2.x = 0
}
```

Exemple en assembleur

```
setup  bis.b  #01h,&P6SEL           ; entrée P6.0 sélectionnée du ADC12
      mov    #REFON+REF2_5V+ADC12ON+SHT0_2,&ADC12CTL0
                                           ; Turn on 2.5V ref, set samp time
      mov    #SHP,&ADC12CTL1        ; Use sampling timer
      mov.b  #SREF_1,&ADC12MCTL0    ; Vr+ = Vref+

      bis.w  #ENC,&ADC12CTL0        ; autorise conversions
startc  bis.w  #ADC12SC,&ADC12CTL0  ; démarre sampling/conversion
test    bit   #BIT0,&ADC12IFG      ; Conversion finie ?
      jz    testIFG                ; Non, on teste encore
      mov   &ADC12MEM0, R5         ; résultat (registre MEM0)
```

Travail personnel

Etudier les éléments référant à l'ADC des fichiers *include 'msp430xxx.h'*.