



aLCiOm



TEXAS INSTRUMENTS



NATIONAL
INSTRUMENTS™

Atelier

Traitement du signal

6 Juin / Après-midi

INSTN – Saclay – 5 & 6 Juin 2012

Agenda : Mercredi 6 juin / Après-midi

14h00 - 14h45 (Alciom)

Exemples d'autres algorithmes fréquemment utiles

14h45 - 15h15 (National Instrument)

Synthèse de l'offre NI pour le traitement du signal

15h15 – 15h30

Pause

15h30 – 16h45 (National Instrument)

L'utilisation de Labview pour le traitement du signal

16h45 – 17h00 (Alciom)

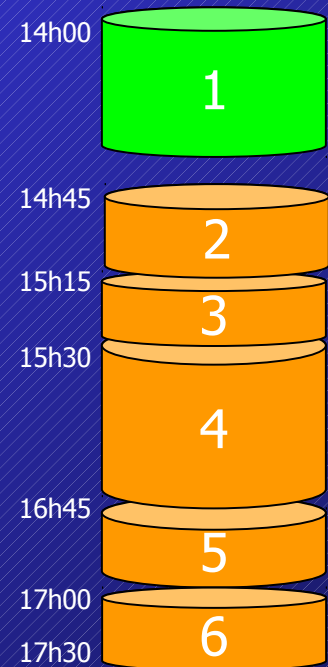
Tendances et synthèse de l'atelier

17h00 - 17h30

Questions-réponses et clôture de la journée

Exemples d'autres algorithmes fréquemment utilisés

Robert Lacoste - ALCIOM

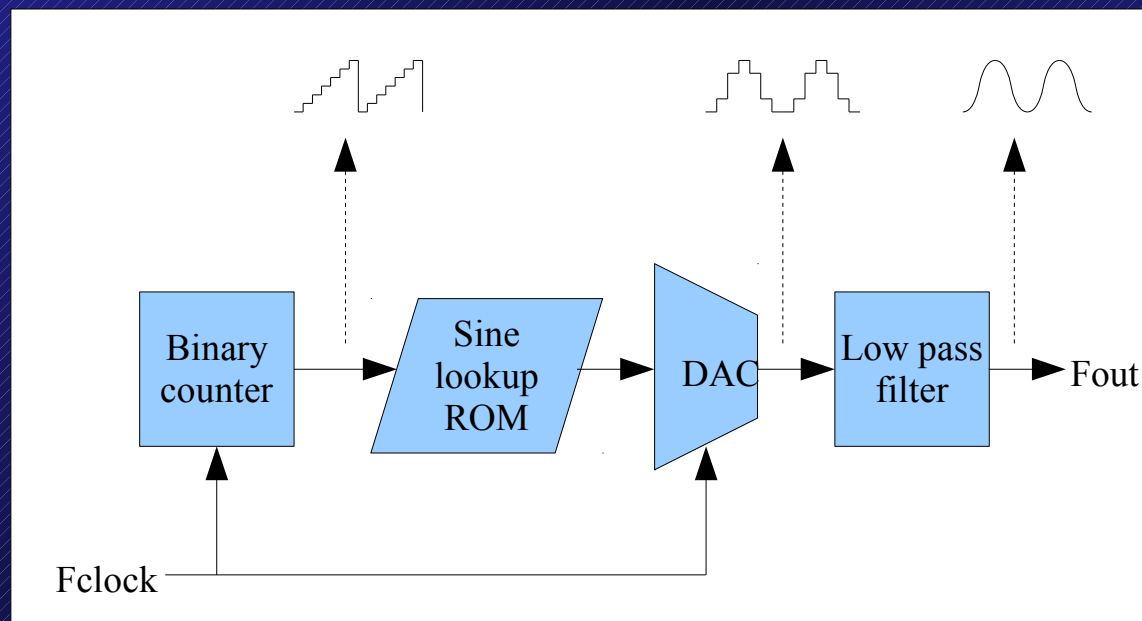


- La génération de signal : introduction aux techniques DDS/NCO
- La cross-corrélation
- L'auto-corrélation
- La détection synchrone

Exemples d'autres algorithmes fréquemment utiles

La génération de signal : introduction aux techniques DDS/NCO

- ◆ Besoin fréquent de générer par logiciel un signal "synthétique"
- ◆ Exemple : générer une sinusoïde via un DAC externe
- ◆ Solution basique : lire à fréquence fixe une table

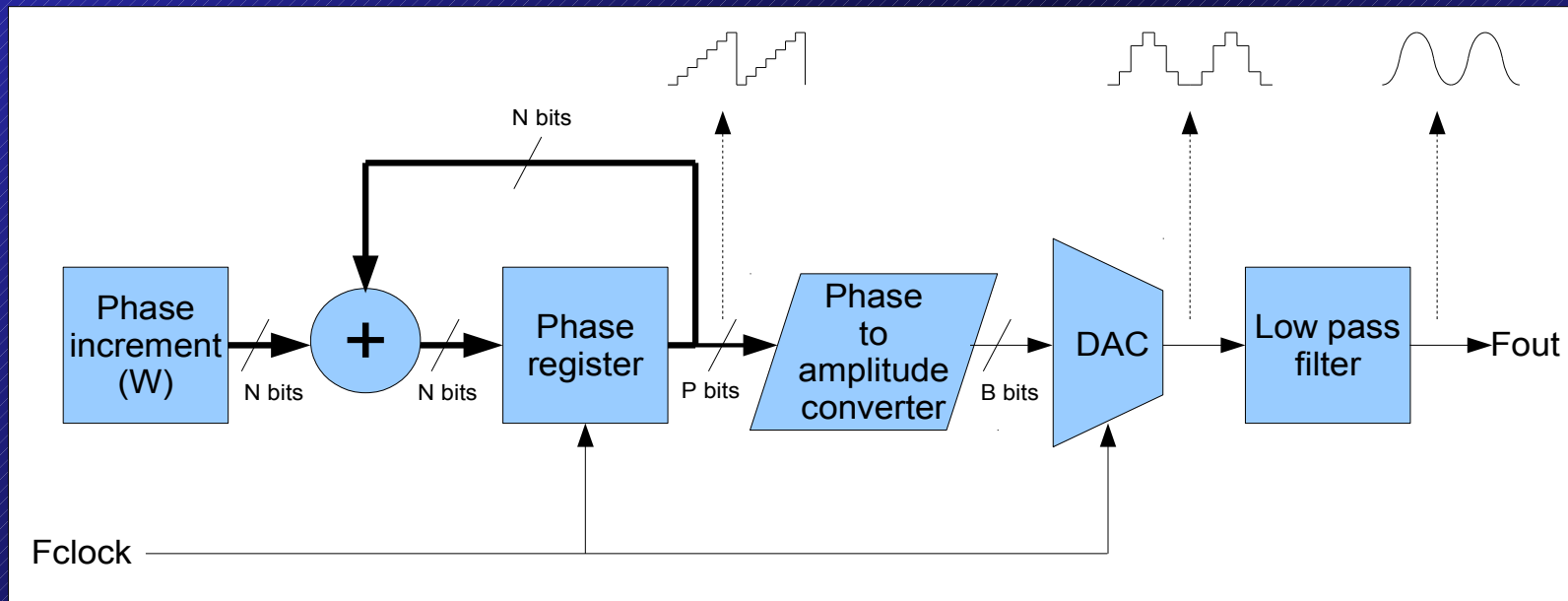


- ◆ Inconvénient : Fréquence générée multiple de 1/période d'un timer, très contraignant en pratique

Exemples d'autres algorithmes fréquemment utiles

La génération de signal : introduction aux techniques DDS/NCO

- ◆ Solution optimisée : DDS (Direct Digital Synthesis)
- ◆ Aussi appelé NCO (Numerically controlled Oscillator)

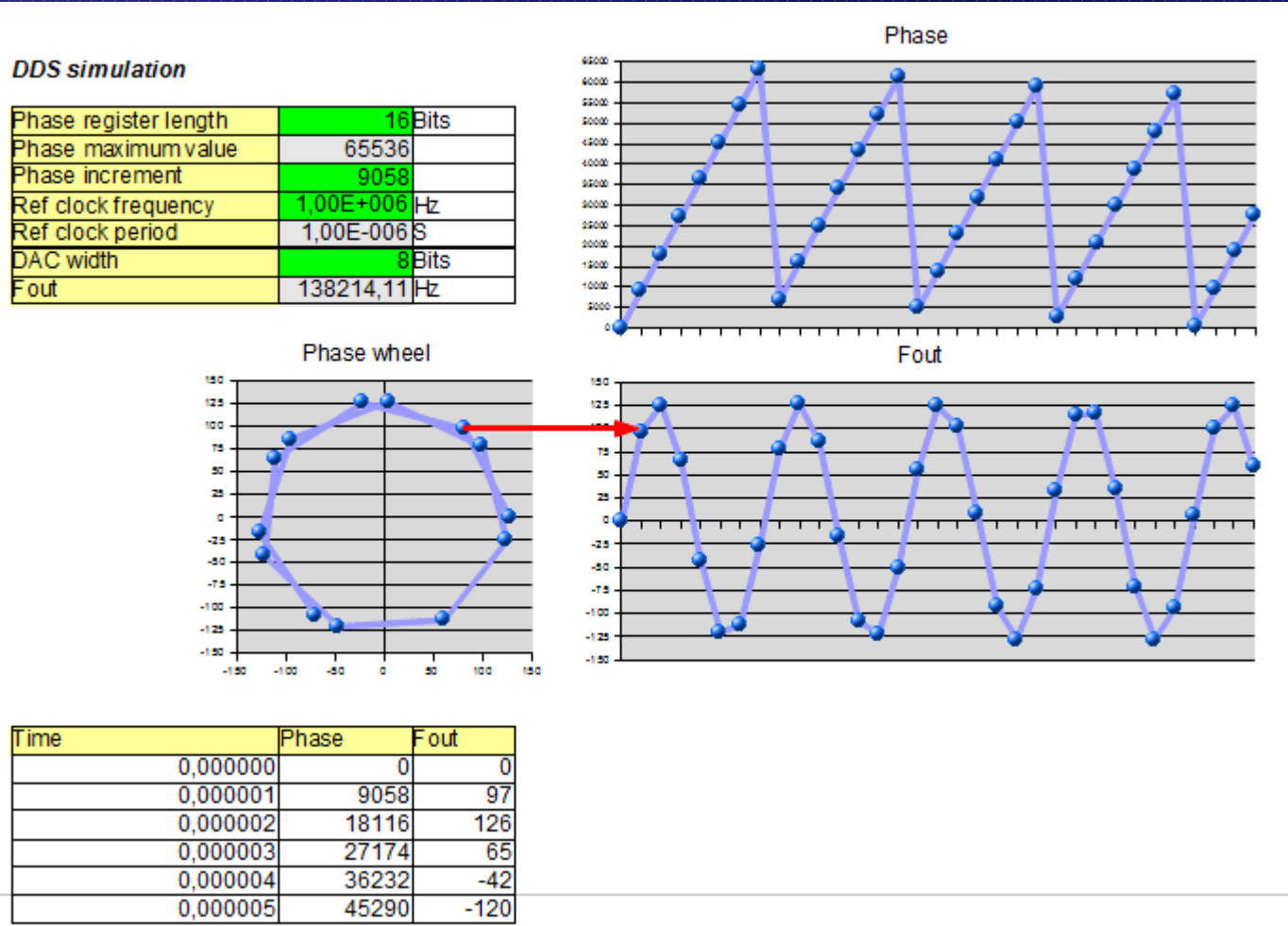


- ◆ La résolution en fréquence n'est plus limitée par la fréquence d'horloge...

Exemples d'autres algorithmes fréquemment utiles

La génération de signal : introduction aux techniques DDS/NCO

◆ Le principe illustré :



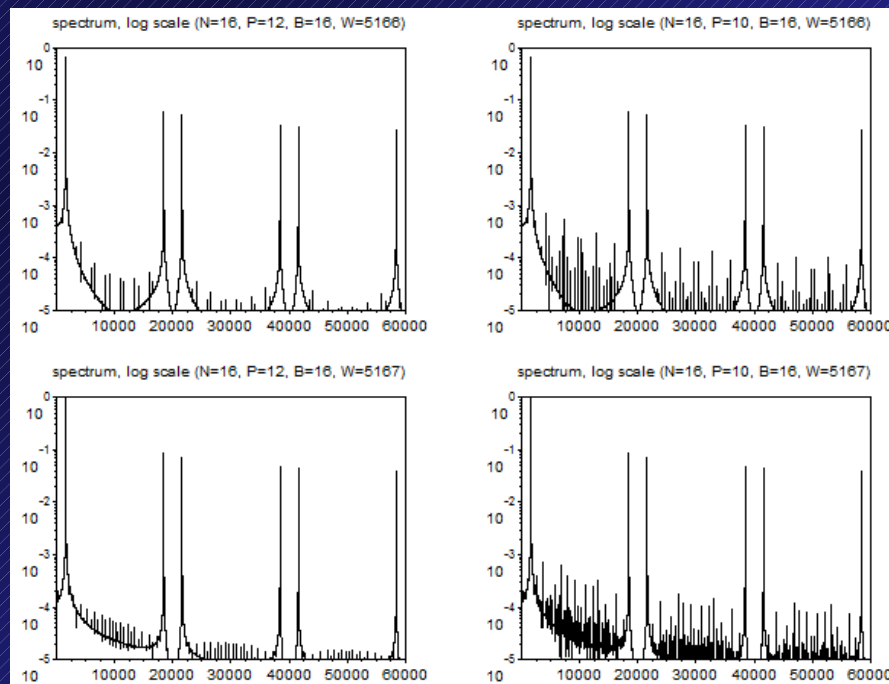
Exemples d'autres algorithmes fréquemment utiles

La génération de signal : introduction aux techniques DDS/NCO

- ◆ Fréquences générables :

$$F_{\text{OUT}} = W \times \frac{F_{\text{CLOCK}}}{2^N}$$

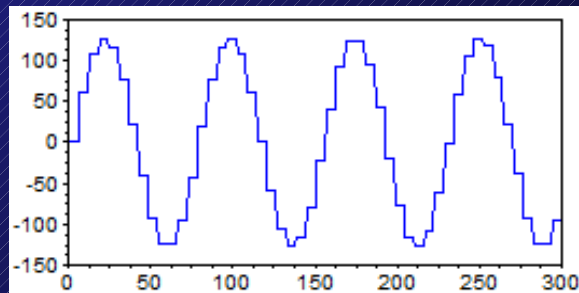
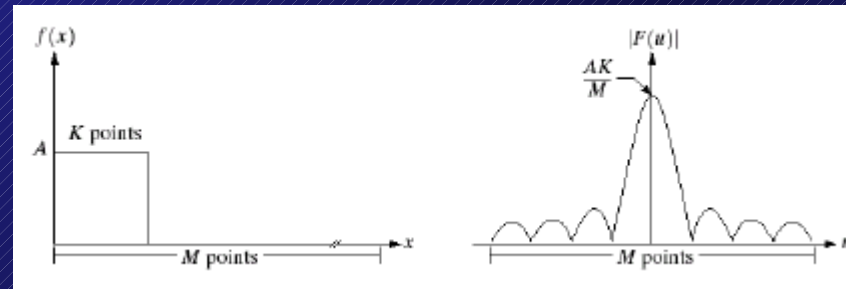
- ◆ Exemple : $F_{\text{clock}}=1\text{MHz}$, $N=32$ bits \rightarrow de 0,00023Hz à presque 500KHz par pas de 0,00023Hz!
- ◆ Filtre passe-bas nécessaire, en particulier si sortie analogique
- ◆ Pureté du spectre fonction de la longueur de la table (P) et de son nombre de bits par mot (B)



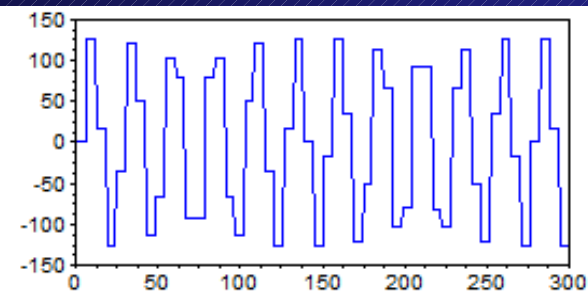
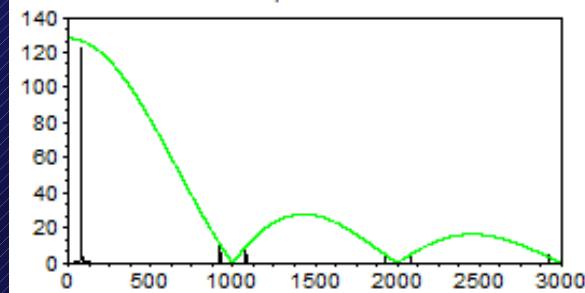
Exemples d'autres algorithmes fréquemment utiles

La génération de signal : introduction aux techniques DDS/NC0

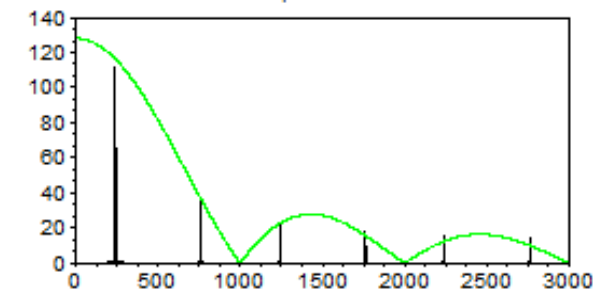
- ♦ Attention : sans compensation le niveau de sortie décroît en fréquence
- ♦ Ce n'est pas une anomalie, mais lié au caractère échantillonné de la sortie
- ♦ Mathématiquement : décroissance en $\sin(x)/x$ (qui est la TF d'un créneau...)
- ♦ -3,92dB à la fréquence de Nyquist



spectrum

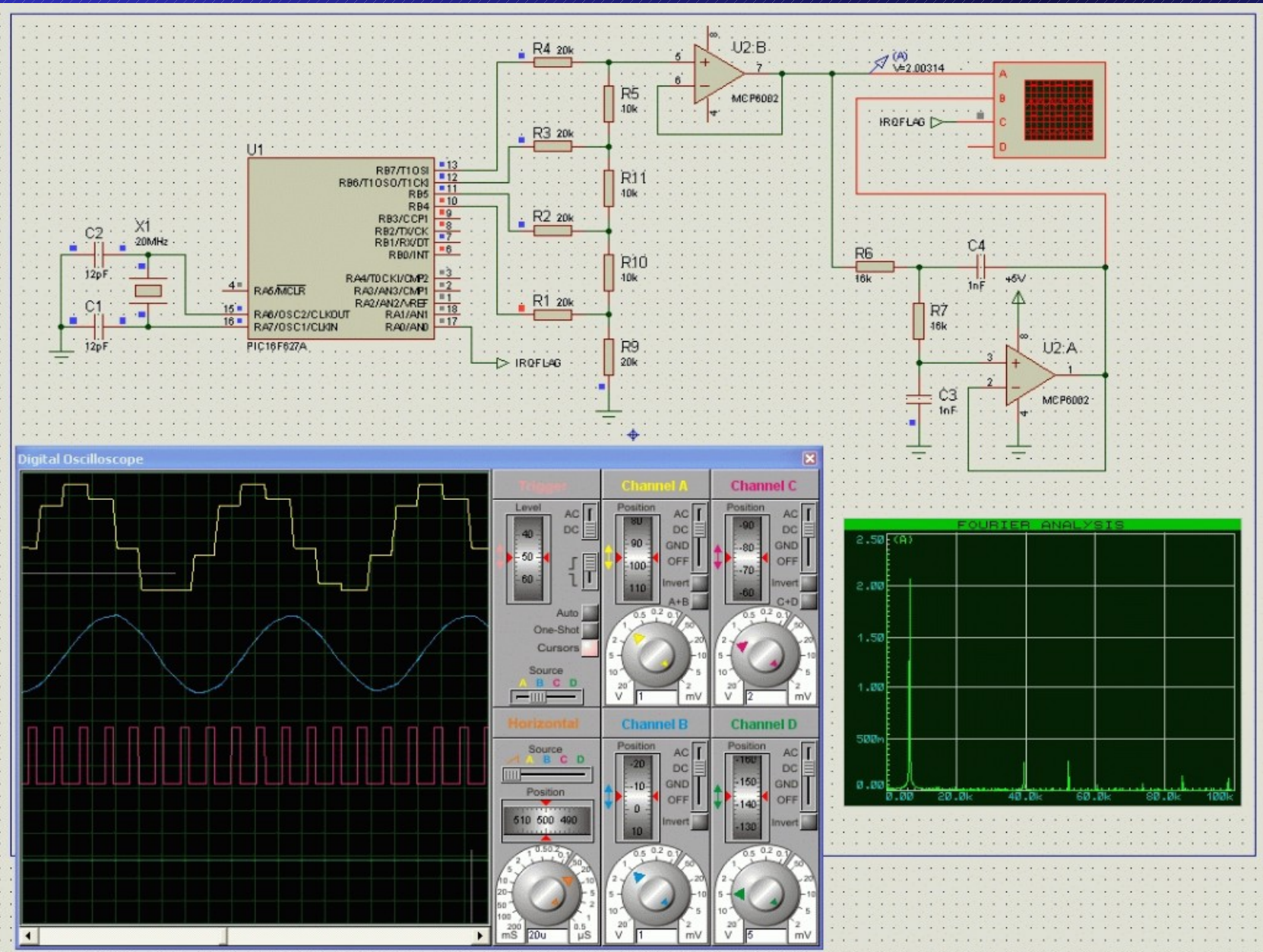


spectrum



Exemples d'autres algorithmes fréquemment utiles

La génération de signal : Exemple de DDS... sur un microcontrôleur 8 bit



Exemples d'autres algorithmes fréquemment utiles

La génération de signal : Exemple de DDS... sur un microcontrôleur 8 bit

```
// Firmware DDS demonstration
// Author   : R.Lacoste for Circuit cellar
// Target   : 16F627A
// Compiler : Hitech PICC-Lite 9.60
// IDE      : MPLAB 8.10

#include    <htc.h>
#define TIMER_VAL (0xFF-100+13)           // Timer reload each 100 cycles (20µs, or 50KHz)
                                           // +13 to compensate interrupt latency

unsigned short phase;                     // Phase register is 16 bits
unsigned short phaseincrement;           // Phase increment is also 16 bits

// Sine table, from 0 to 255
const unsigned char SINE_TABLE[]=
{
128, 131, 134, 137, 140, 143, 146, 149, 152, 156, 159, 162, 165, 168, 171, 174,
176, 179, 182, 185, 188, 191, 193, 196, 199, 201, 204, 206, 209, 211, 213, 216,
218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 237, 239, 240, 242, 243, 245,
246, 247, 248, 249, 250, 251, 252, 252, 253, 254, 254, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 254, 254, 253, 252, 252, 251, 250, 249, 248, 247,
246, 245, 243, 242, 240, 239, 237, 236, 234, 232, 230, 228, 226, 224, 222, 220,
218, 216, 213, 211, 209, 206, 204, 201, 199, 196, 193, 191, 188, 185, 182, 179,
176, 174, 171, 168, 165, 162, 159, 156, 152, 149, 146, 143, 140, 137, 134, 131,
127, 124, 121, 118, 115, 112, 109, 106, 103, 99, 96, 93, 90, 87, 84, 81,
79, 76, 73, 70, 67, 64, 62, 59, 56, 54, 51, 49, 46, 44, 42, 39,
37, 35, 33, 31, 29, 27, 25, 23, 21, 19, 18, 16, 15, 13, 12, 10,
9, 8, 7, 6, 5, 4, 3, 3, 2, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 2, 3, 3, 4, 5, 6, 7, 8,
9, 10, 12, 13, 15, 16, 18, 19, 21, 23, 25, 27, 29, 31, 33, 35,
37, 39, 42, 44, 46, 49, 51, 54, 56, 59, 62, 64, 67, 70, 73, 76,
79, 81, 84, 87, 90, 93, 96, 99, 103, 106, 109, 112, 115, 118, 121, 124
};
```

Exemples d'autres algorithmes fréquemment utiles

La génération de signal : Exemple de DDS... sur un microcontrôleur 8 bit

```
// timer 0 interrupt routine
void interrupt timer0_isr(void)
{
    unsigned char lookupindex;

    TMR0=TIMER_VAL;           // Reload timer
    PORTA=0b00001;           // Signal interrupt processing on RA0
    phase+=phaseincrement;    // Increment the phase register
    lookupindex=(unsigned char)(phase>>8); // get the top 8 bits
    PORTB=SINE_TABLE[lookupindex]; // and convert it to a sine
    PORTA=0b00000;
    TOIF = 0;                // clear interrupt flag
}

// Main program
main()
{
    // initialize DDS
    phase=0;
    phaseincrement= 9328; // Fout= 50KHz x phaseincrement/65536 = 7116,69Hz

    // initialize timer 0
    OPTION = 0b1000; // timer mode, no prescaler
    TOCS = 0;        // select internal clock
    TMR0=TIMER_VAL;  // Initialize timer
    TOIE = 1;        // enable timer interrupt
    GIE = 1;         // enable global interrupts
    TRISB = 0;       // port B is output
    TRISA = 0b11110; // RA0 is output
    PORTB = 0;       // Port B start at 0

    for(;;)
        continue; // Do nothing, all is done in the interrupt
}

// That's all, folks.
```

Exemples d'autres algorithmes fréquemment utiles

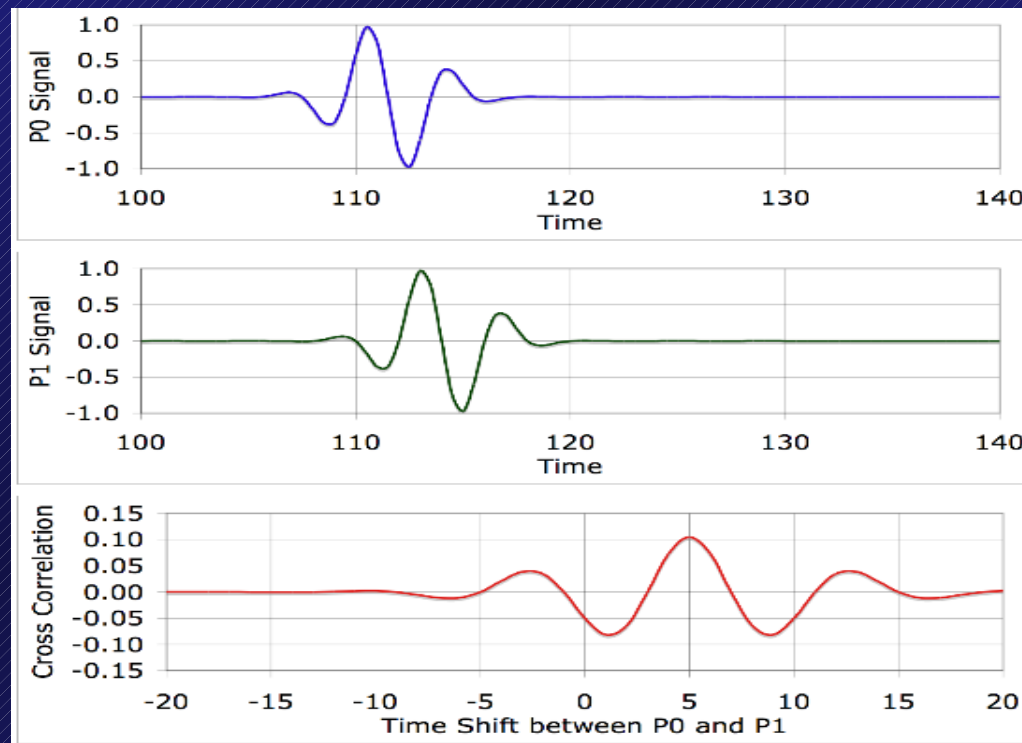
La cross-corrélation

- ◆ Recherche de « correspondance » entre deux signaux numérisés :

→ Calcul de cross-corrélation

$$(f \star g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m] g[n+m]$$

- ◆ Mesure la correspondance de deux signaux avec délai variable entre les deux
- ◆ Revient à mesurer l'amplitude de la convolution des deux signaux
- ◆ Convolution donc algorithme rapide avec des FFT...



Exemples d'autres algorithmes fréquemment utiles

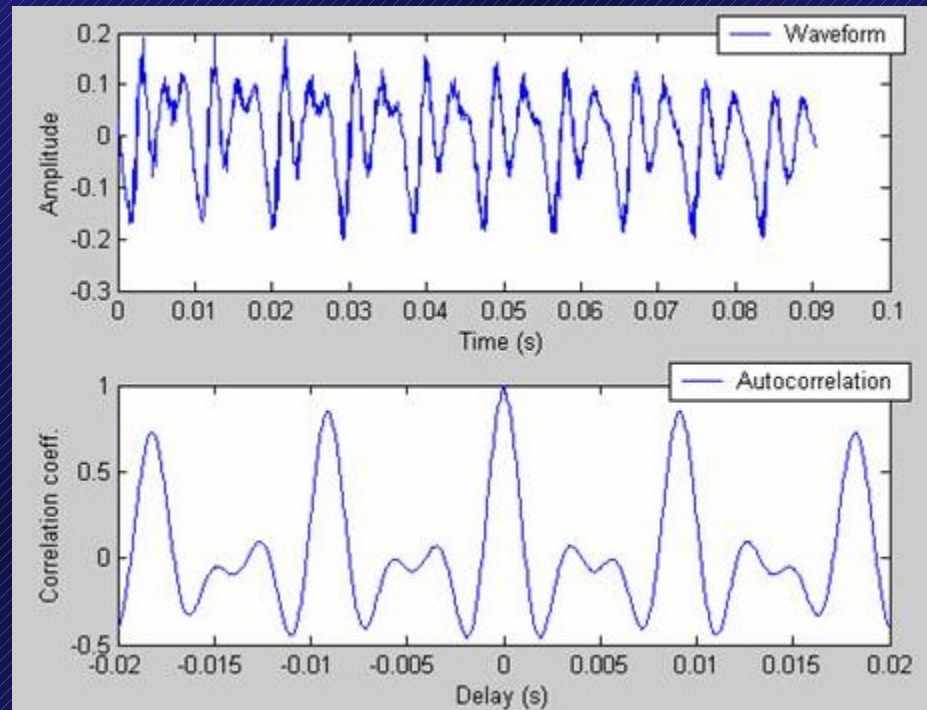
L'auto-corrélation

- ◆ Recherche de « périodicité d'un signal » :

→ Calcul d'autocorrélation

$$R_{xx}(j) = \sum_n x_n \bar{x}_{n-j}$$

- ◆ Revient à calculer la cross-corrélation du signal avec lui-même



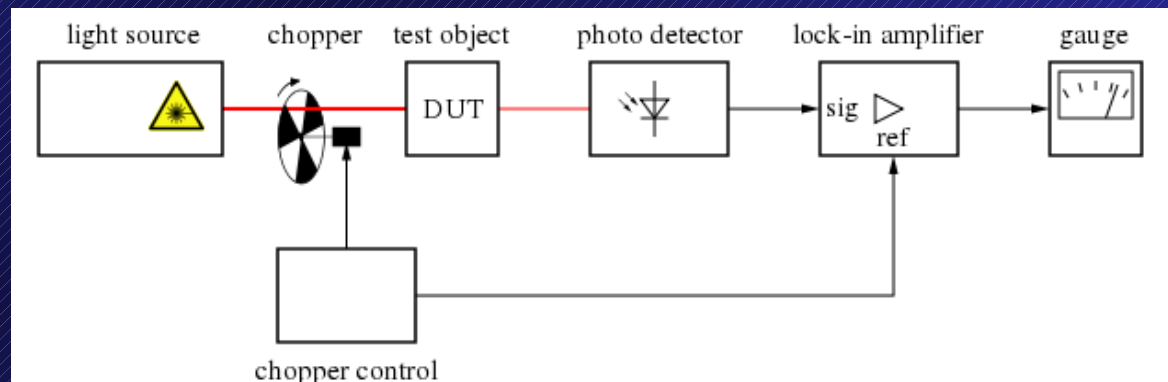
Exemples d'autres algorithmes fréquemment utiles

La détection synchrone

Un cas spécifique de cross-corrélation : la détection synchrone (lock-in amplifier)

Principe :

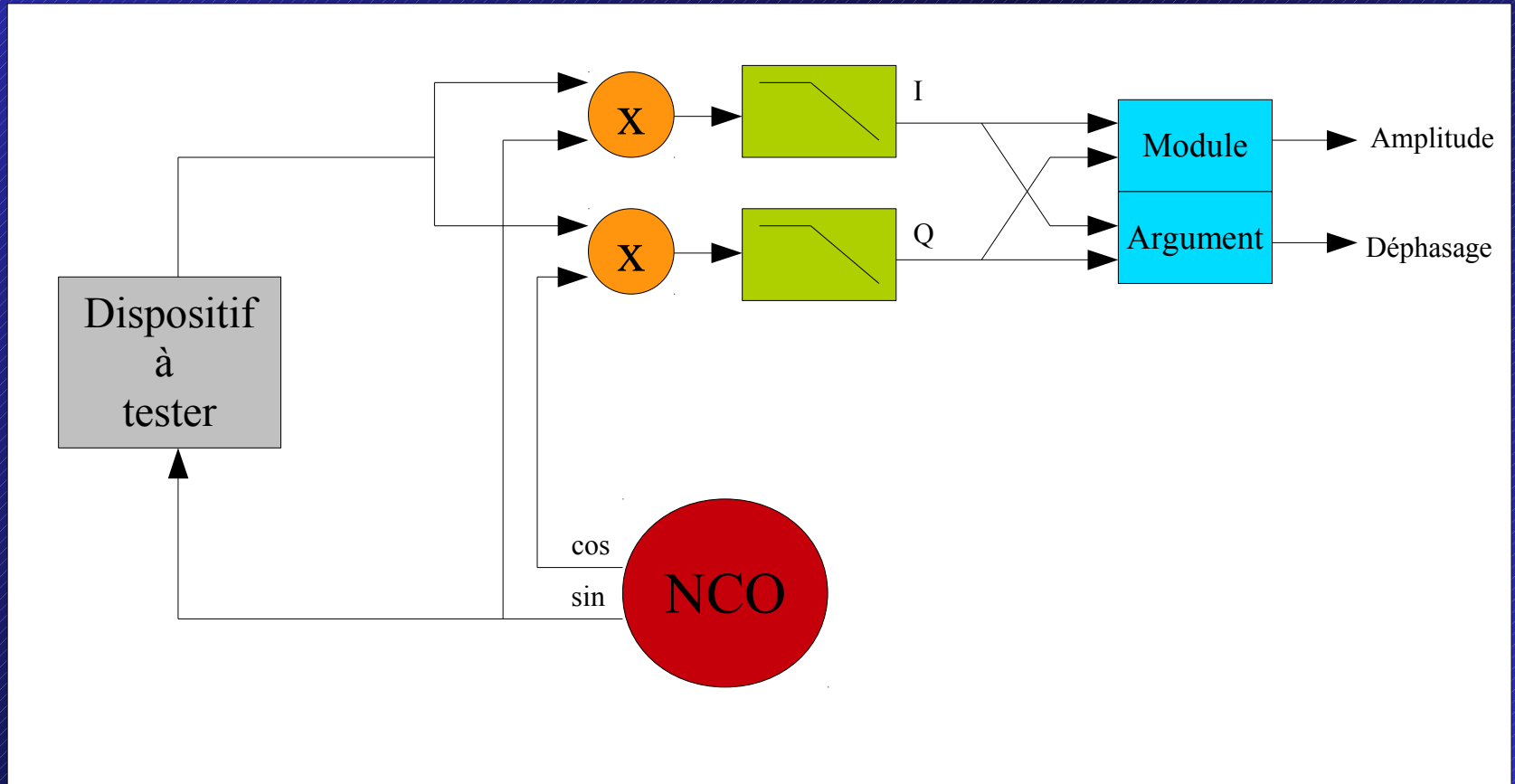
- ♦ Génération d'un signal d'excitation périodique (en pratique sinusoïdal)
- ♦ Détection du signal reçu par cross-corrélation avec le signal d'excitation, connu
- ♦ Permet une très grande insensibilité au bruit, car le gain du filtrage peut être très grand (limité par la durée de mesure)
- ♦ Exemple : Mesure de signaux 100dB sous le bruit...



Exemples d'autres algorithmes fréquemment utiles

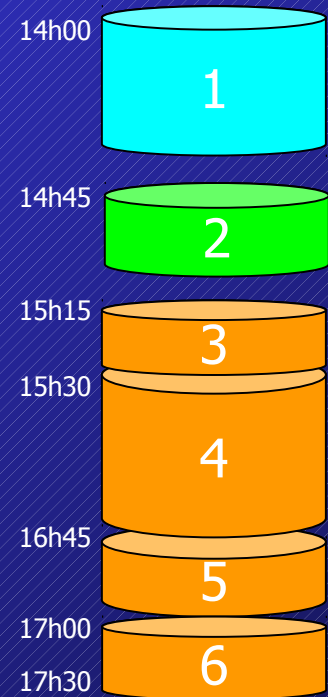
La détection synchrone

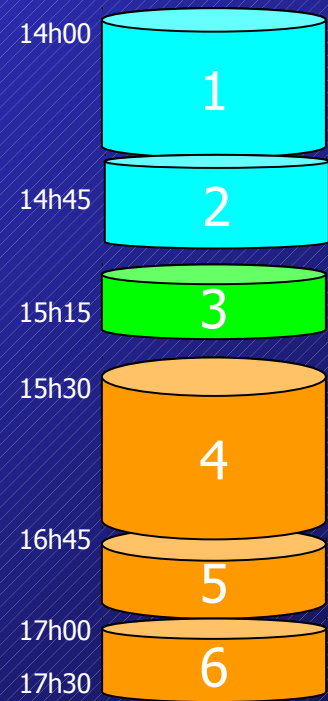
- ◆ Un exemple d'architecture, basé sur une démodulation I/Q



Synthèse de l'offre NI pour le traitement du signal

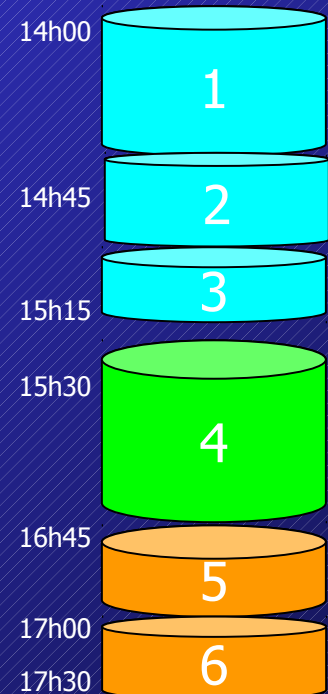
Emmanuel Roset – National Instruments





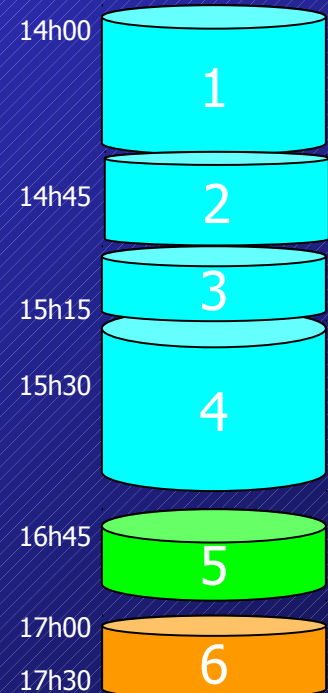
Partie 4 : L'utilisation de Labview pour le traitement du signal

Emmanuel Roset – National Instruments



Tendances et synthèse de l'atelier

Robert Lacoste - ALCIOM

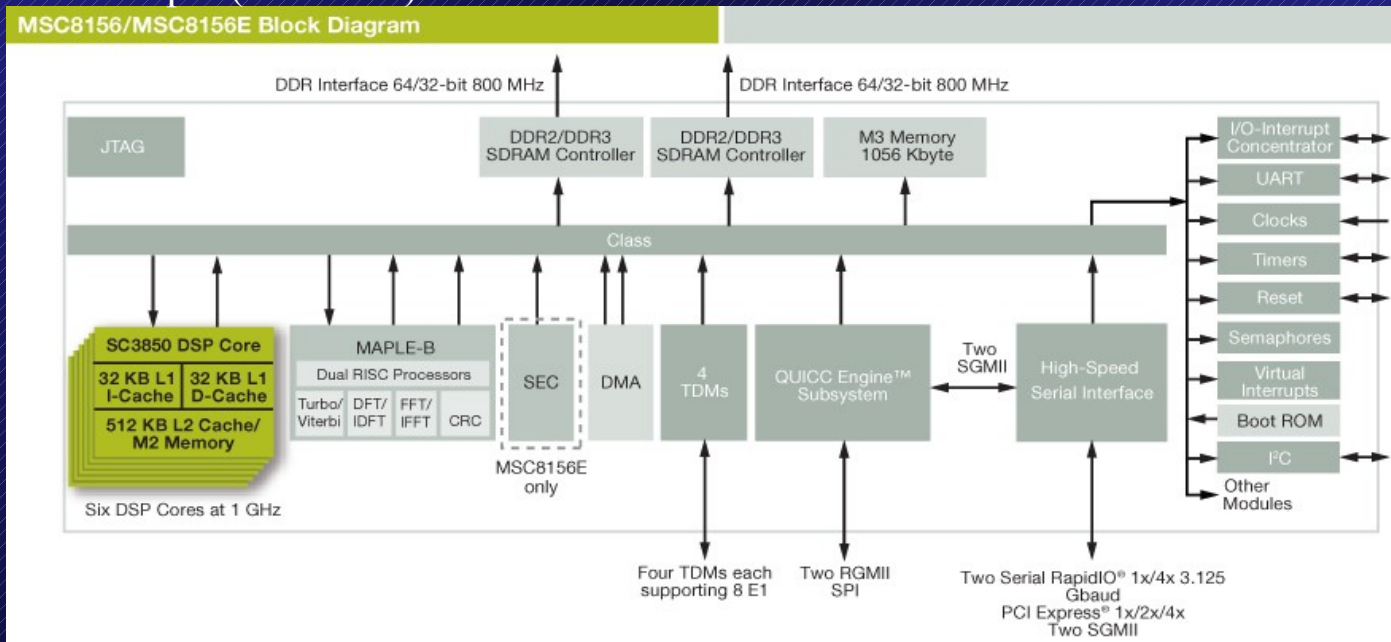


- Les grandes tendances technologiques
- Traitement du signal et pluridisplinarité des équipes
- Quelques recommandations méthodologiques
- Les grandes erreurs à éviter
- Les grandes réussites
- Quelques références

Les grandes tendances technologiques

- ◆ Le traitement numérique du signal n'est PLUS une technologie de pointe !
- ◆ Traitement sur microcontrôleurs maintenant souvent envisageable (ARM etc)
- ◆ " micro-DSP " très économiques et simples de mise en oeuvre
- ◆ Pour les applications plus exigeantes, combos CPU+DSP disponibles
- ◆ Consommations énergétiques de mieux en mieux optimisées
- ◆ Multi-coeurs, génériques ou souvent spécialisés...

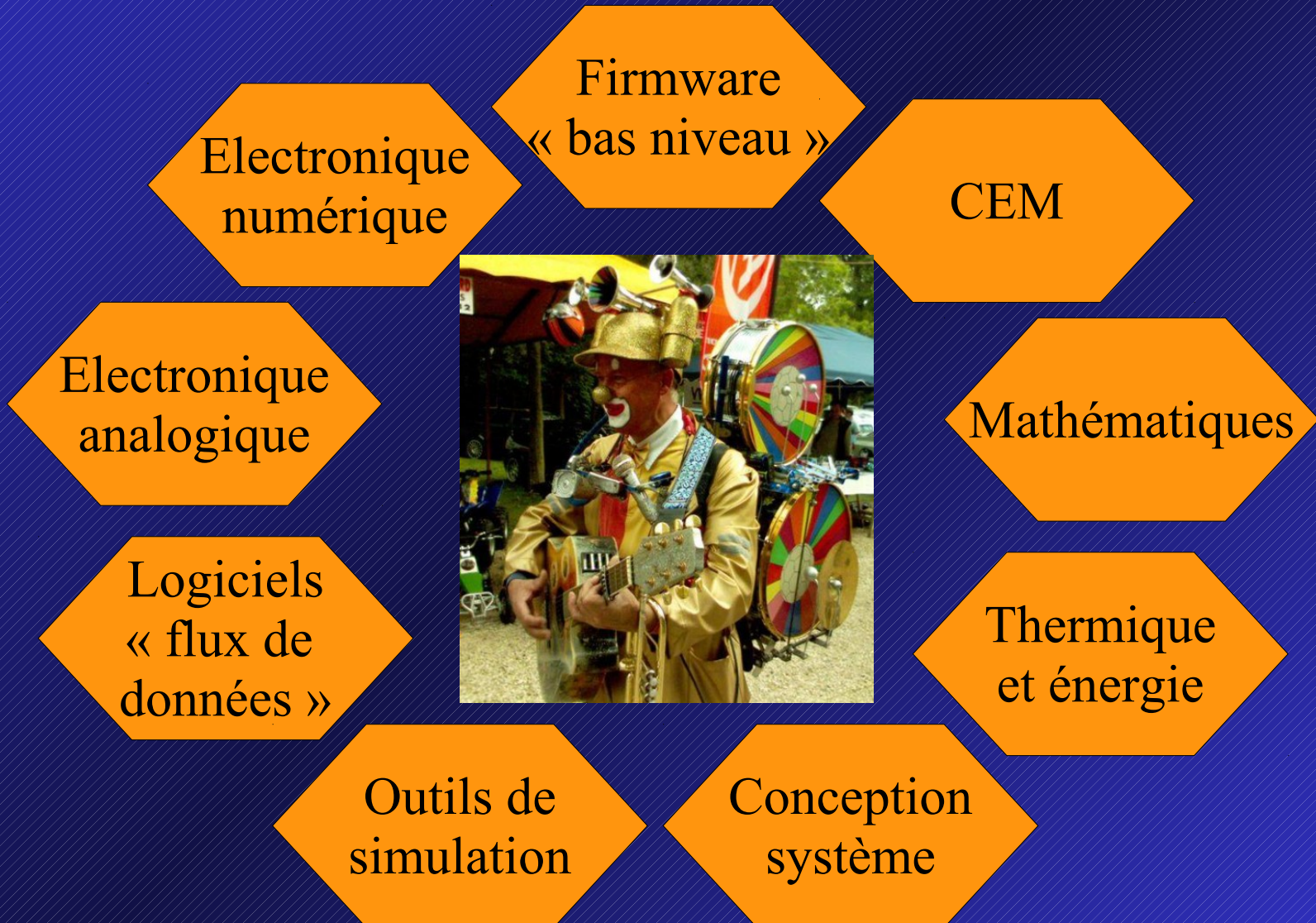
Un exemple (Freescale) :



Clôture

Traitement du signal et pluridisciplinarité des équipes

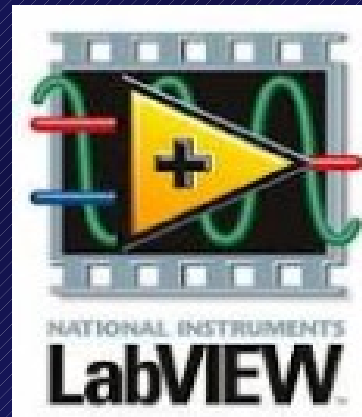
- ♦ Souvent la plus grosse difficulté pour se « lancer » dans le traitement du signal...



Clôture

Quelques recommandations méthodologiques

- ♦ Bien réfléchir au besoin avant de se lancer...
- ♦ Quelle fréquence d'acquisition minimale ? Possibilité de pré-traiter en analogique ?
- ♦ Comment réduire le plus rapidement possible le flux de données ?
- ♦ **Simuler** le plus tôt possible !!!
- ♦ Numériser dès que possible des signaux réels, et les traiter en simulation



Clôture

Quelques recommandations méthodologiques

- ◆ Ne pas réinventer la lune...
- ◆ Penser aux bibliothèques disponibles, et déjà optimisées...

TMS320C6000 DSP Library

SPRC265

TMS320C67x DSP Library

SPRC121

BLACKFIN SOFTWARE MODULES

CMSIS_DSP_HEADERS:

Cortex Microcontroller Software

dsPIC DSC DSP Algorithm Library

Part Number: SW300022

PIC32 DSP Library



SigLib™

Supported Processors

- TMS320 C3x, C4x, C54x, C6000™
- Pentium™ / SPARC™ / PowerPC™
- ADSP 21xx 21xxx™
- StarCore 810x™
- ARM, XScale

INTEL® SOFTWARE DEVELOPMENT PRODUCTS	Home	Product
--------------------------------------	------	---------

Multicore Power for Multimedia and Data Processing

Intel® Integrated Performance Primitives (Intel® IPP) 7.0

Clôture

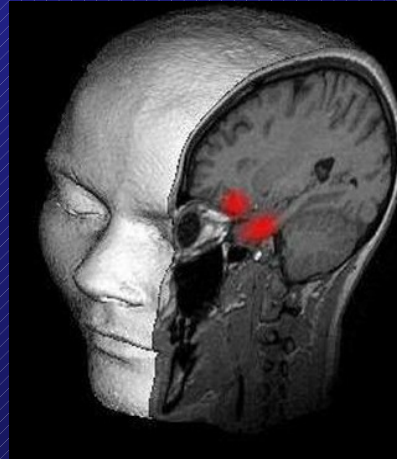
Les grandes erreurs à éviter

- ♦ Ne pas écrire les spécifications du produit produit avant de penser DSP
- ♦ Ne pas se limiter dans un premier temps aux seules fonctionnalités essentielles...
- ♦ Tester ses algorithmes avec des signaux parfaits...
- ♦ Utiliser un algorithme « trouvé sur le web » sans comprendre ce qu'il fait...
- ♦ Développer un prototype avec le processeur de puissance « juste suffisante »...
- ♦ Ne pas calculer la quantité de mémoire nécessaire (RAM surtout)
- ♦ Economiser lors de l'achat des outils de développement et de mise au point
- ♦ Economiser quelques milliers d'euros pour ne pas acheter une librairie DSP
- ♦ Ne pas acheter le kit d'évaluation...
- ♦ Ne pas prévoir un moyen pour reprogrammer le code in situ...
- ♦ Ne pas réfléchir à la consommation électrique... et à la CEM
- ♦ Ne pas former vos équipes...
- ♦ Ne pas travailler en équipe...
- ♦ Ne pas vous faire aider...



Clôture

Les grandes réussites

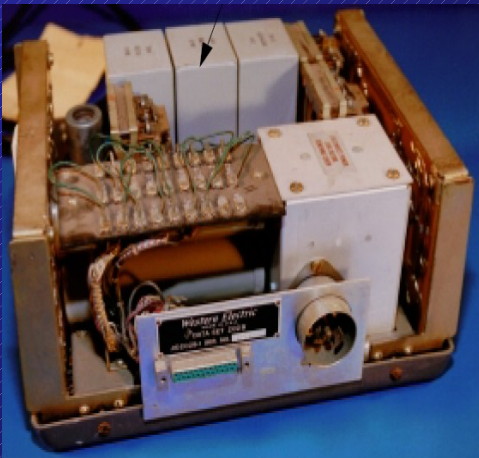


Clôture

Les grandes réussites



2010 : 24 Mbps



The Bell 202 FSK, 1200 b/s, Four-wire Leased-line Modem

1950 : 1200 bps



Codex 3600 Modem that Used a Higher Symbol Rate and Trellis Precoding to Attain Transmission Rates of 24000 b/s

1990 : 24000 bps

20 ans, x1000

40 ans, x20

Clôture
Et...



A vous de jouer!

Clôture

Quelques références

Understanding digital signal processing

Richard G. Lyons, Prentice hall, ISBN 978-0-13-702741-5

Analog Circuits – World class designs

Robert A. Pease, Newnes, ISBN 978-0-7506-8627-3

The Darker Side

Robert Lacoste, Newnes, ISBN 978-1-85617-762-7

"A technical tutorial on Digital Signal Synthesis", Analog Devices 1999,

http://www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99.pdf

The Data Conversion Handbook, Walt Kester (Analog Device), March 2004

http://www.analog.com/library/analogDialogue/archives/39-06/data_conversion_handbook.html

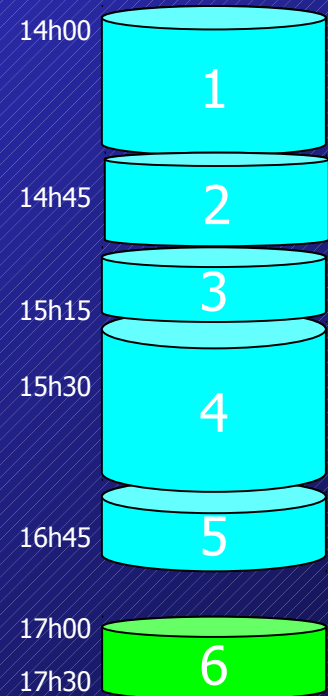
Playing with high speed ADC

Robert Lacoste, Circuit Cellar 259, February 2012

Processeurs DSP, Olivier Sentieys, IRISA

<http://www2.lirmm.fr/archi07/archi07-sentieys.pdf>

Questions / Réponses et clôture de l'atelier





Experts en signaux mixtes

5, parvis Robert Schuman
92370 Chaville
www.alciom.com

Robert Lacoste
Tel : 01 47 09 30 51
Mail : rlacoste@alciom.com