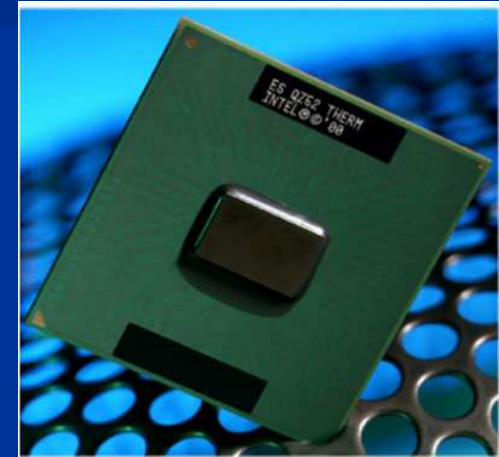
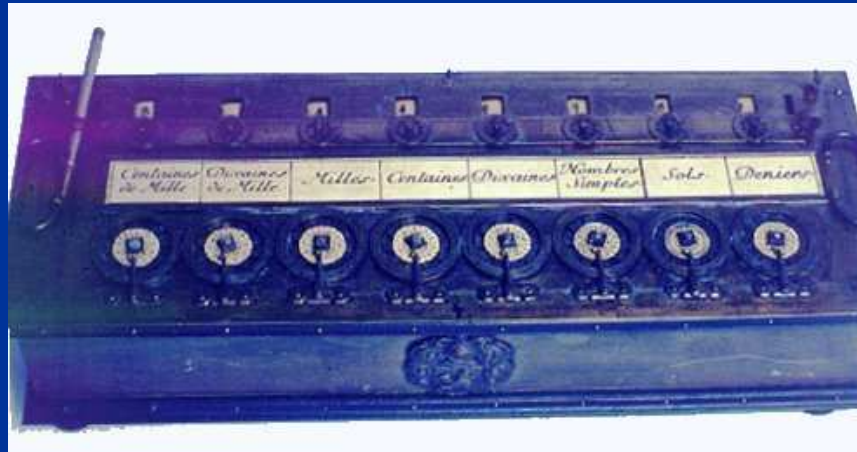


Microcontrôleurs

Version PIC

Le monde du BIT de la
microseconde et du nanomètre



Christian Dupaty
Académie d'Aix-Marseille
christian.dupaty@ac-aix-marseille.fr

Objectifs



- Être capable de concevoir un programme simple en langage C pour microcontrôleur Microchip PIC
- Analyse, algorithme, programmation structurée, jeux d'instructions, modes d'adressage
- Mise en œuvre de l'outil de développement Microchip MPLAB+ PICDEM2+

Microprocesseurs VS Microcontrôleurs

- Microprocesseurs, ex : PENTIUM, AMD, 68000

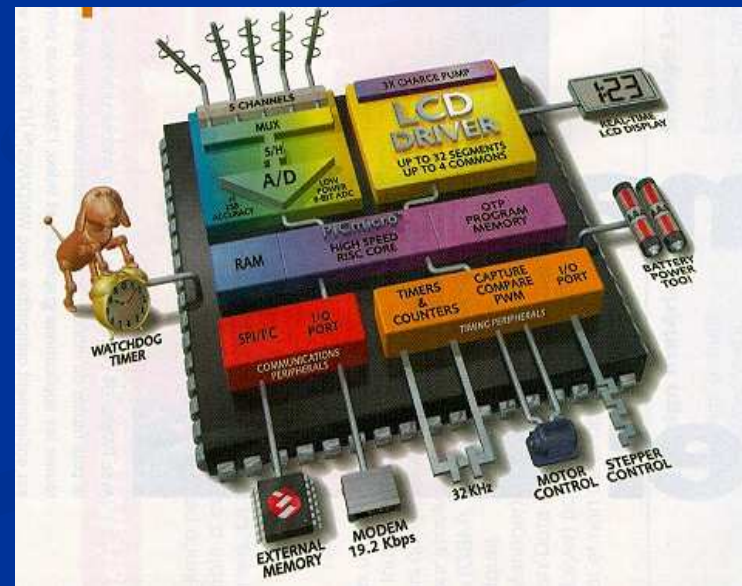


- Microcontrôleurs , ex : 8051, 68HC11, PIC18, AVR, ST6 ...



Microcontrôleurs - Avantages

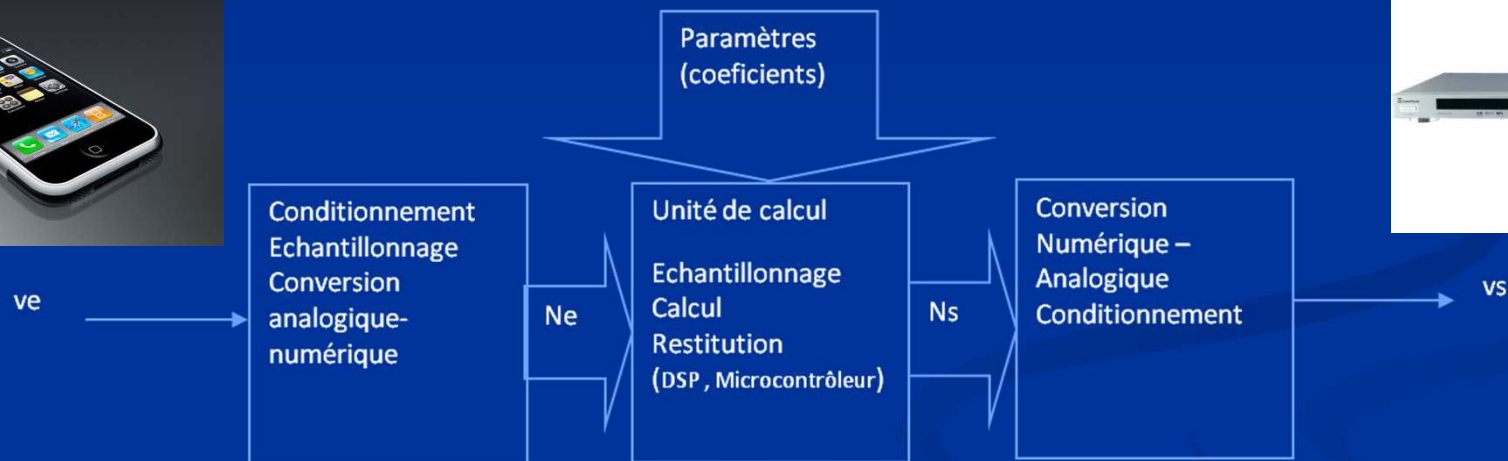
- Faible cout
- Reconfigurable (modification du programme)
- Fiabilité, tout est sur la même puce (processeur, mémoires, périphériques)
- Faible consommation



Domaines d'emplois: Petits automatismes



Domaines d'emplois: Traitement numérique du signal (Digital Signal Processor)

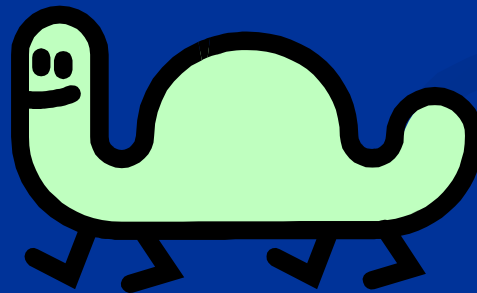


Caractéristiques uP/uC

	Petits automatismes	Informatique industrielle
Fréquences	4 à 40MHz	400MHz à 4.0GHz
Puissance de calcul	faible	Très élevé
Mémoire vive	RAM : 1KO	RAM : 4GO ROM : 1MO
Mémoire programme	ROM FLASH: 64 KO	Disque dur
Intégration des périphériques	Maximum	minimum
consommation	faible	Elevé
prix	2€ à 50€.	200€ à 1000€
Composant utilisé	Microcontrôleur 8/16 bits	Microprocesseur 32/64 bits

Éléments de numération binaire

Les origines des calculateurs



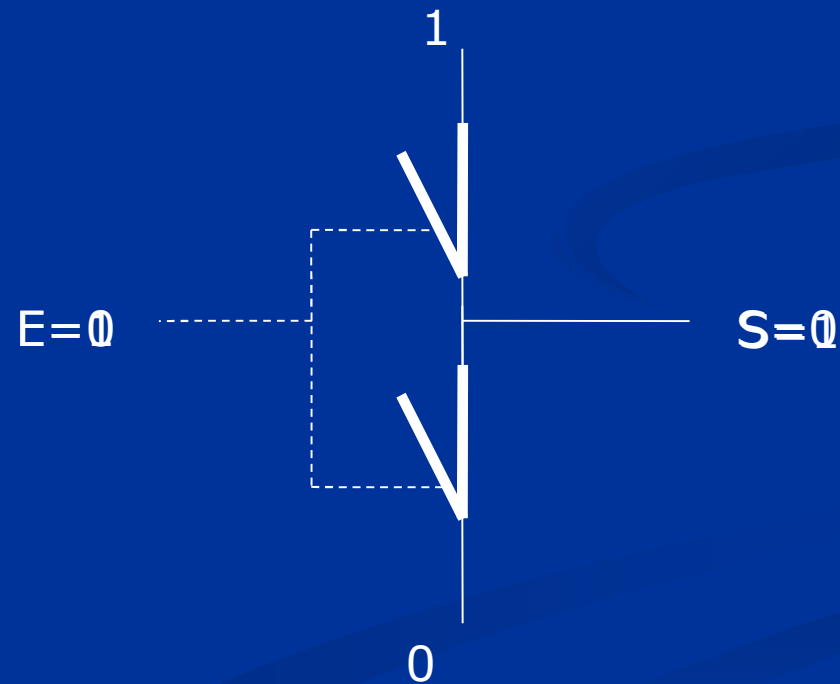
0110010011011011110101100010010001010110001001011000111100101

Le BIT : 0 ou 1

- En électronique numérique toutes les opérations s'effectue en base 2
- Aux deux états logiques sont associés deux tensions :
 - En logique « positive » 0 → 0v et 1 → 1v à 5v
 - En logique « négative » 1 → 0v et 0 → 1v à 5v

Interrupteur électronique

- Ces deux états électriques dépendent de la conduction ou du blocage d'un composant électronique
- Ex : Fonction Booléenne NON, l'inverseur



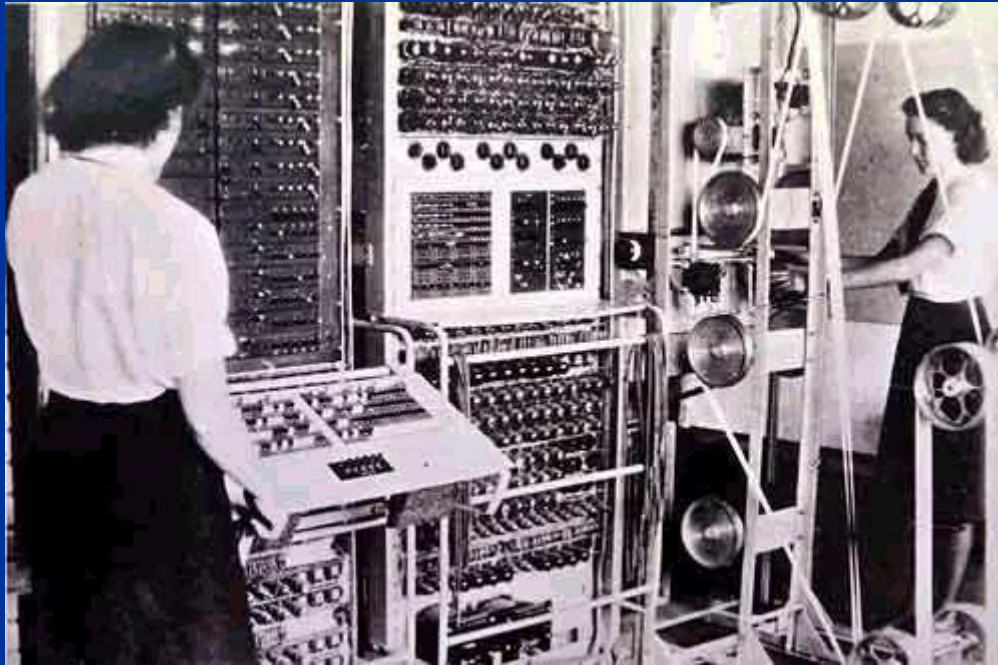
AVANT

- TUBE TRIODE (Lee De Forest en 1906.)

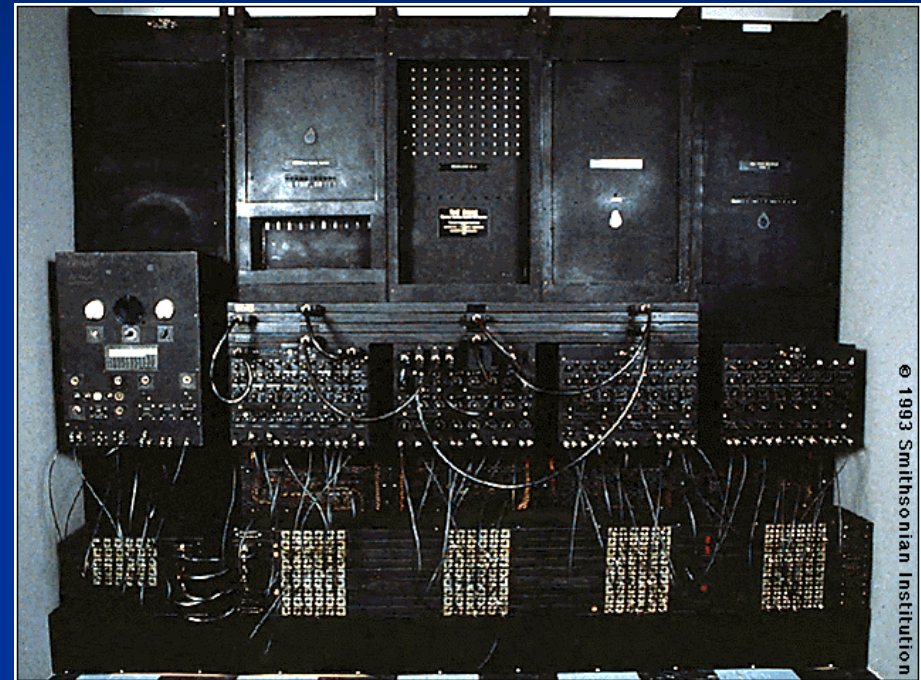
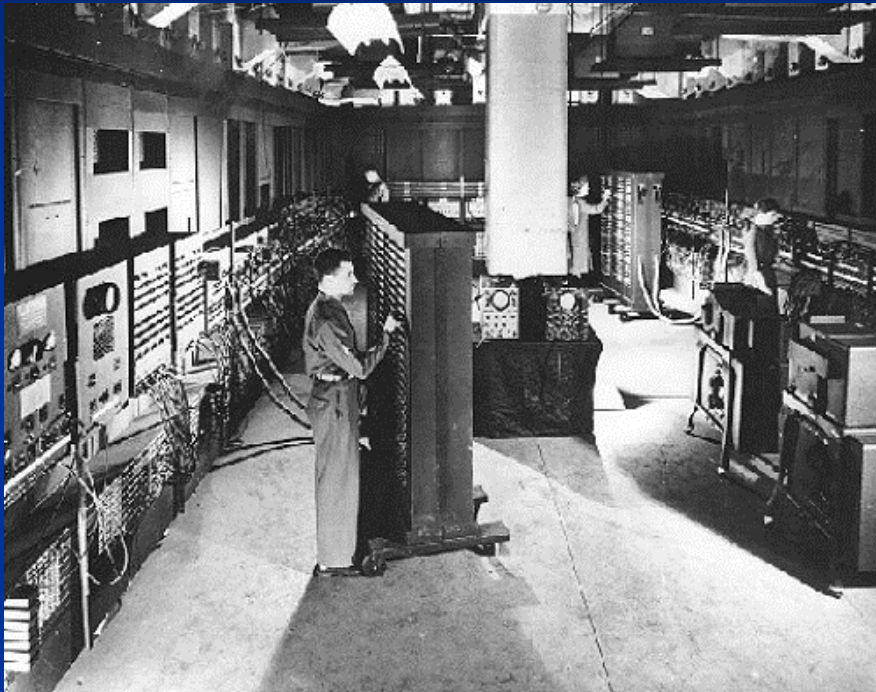


1940 Z3, COLOSSUS

Inventés par Konrad Zuse et Arnold Lynch
Ordinateurs électromécaniques (commutateurs à relais)
fréquence du processeur : 5,33 Hz
puissance réelle : 20 Flops



Premier Ordinateur ELECTRONIQUE– ENIAC 1946



17 468 tubes à vide, 7 200 diodes à cristal, 1 500 relais, 70 000 résistances, 10 000 condensateurs et environ 5 millions de joints soudés à la main. Son poids est de 30 tonnes.

5 000 additions simples ou 357 multiplications ou 38 divisions par seconde.

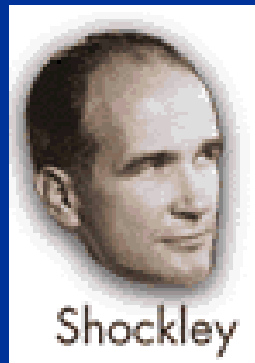
L'invention du Transistor 1947



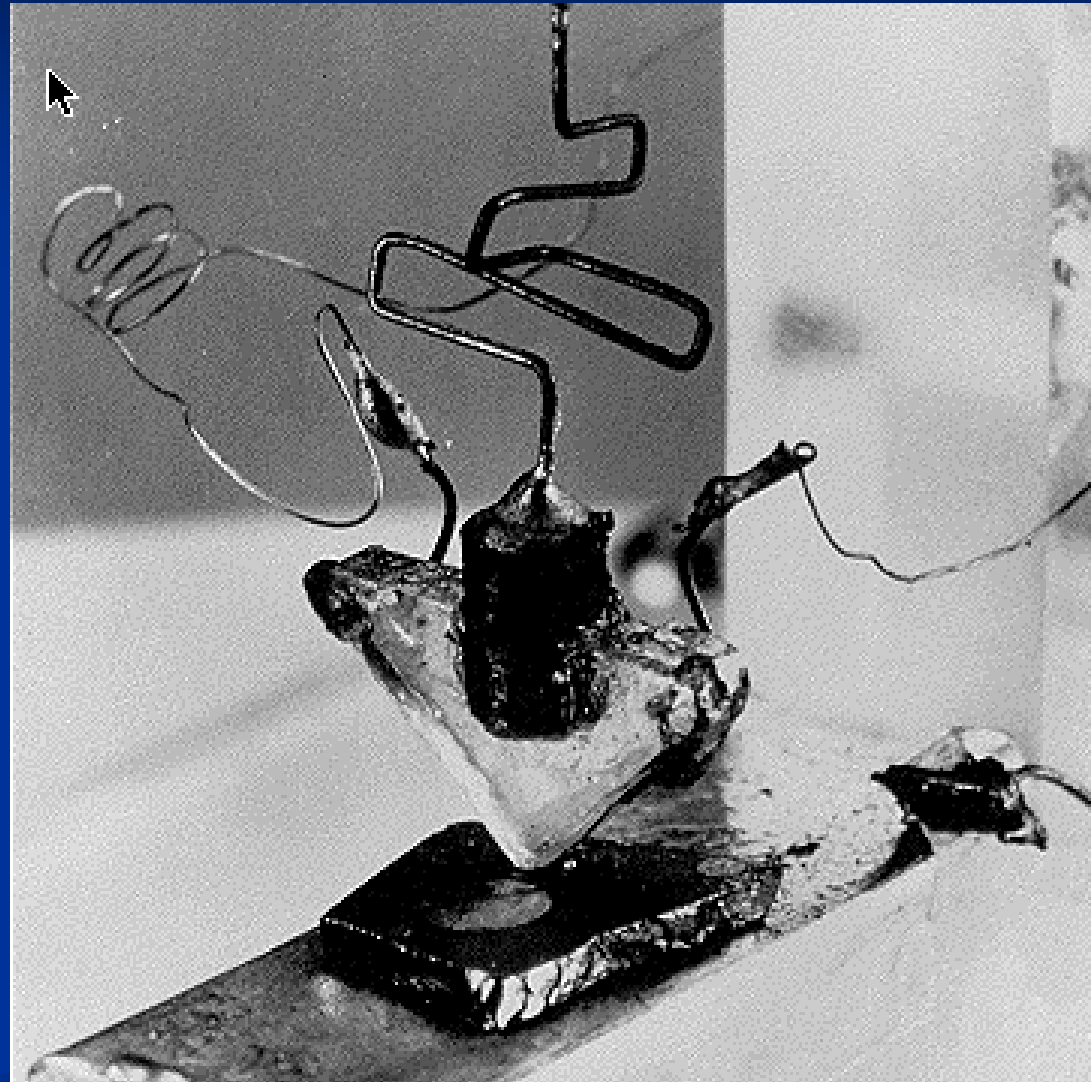
Brattain



Bardeen

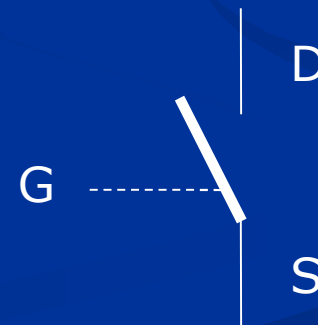
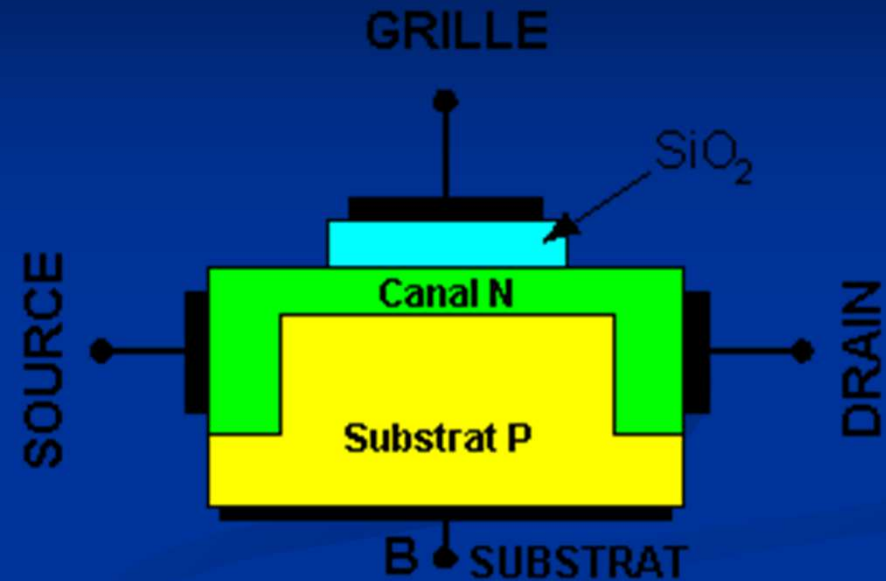


Shockley

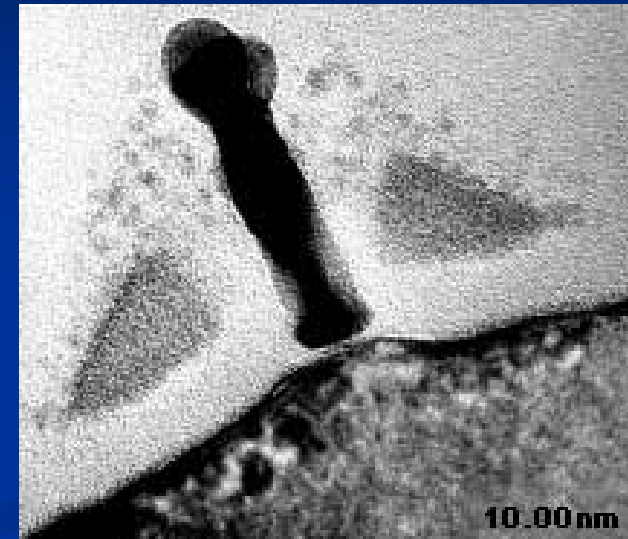
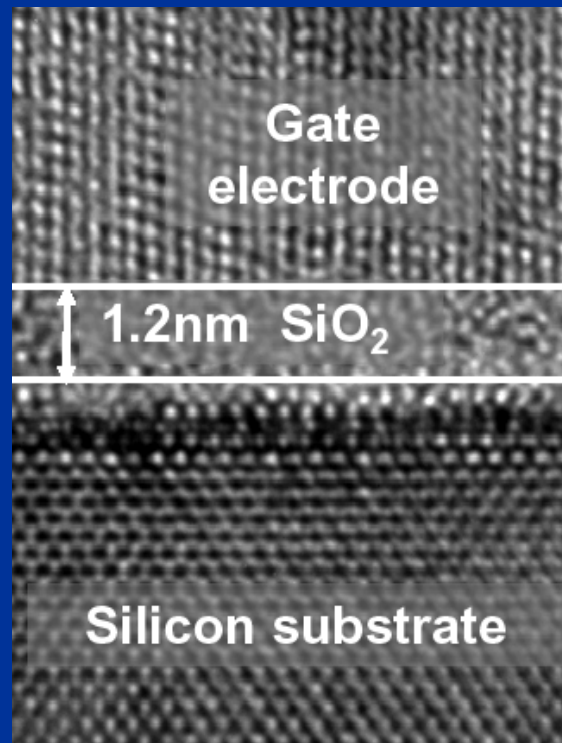
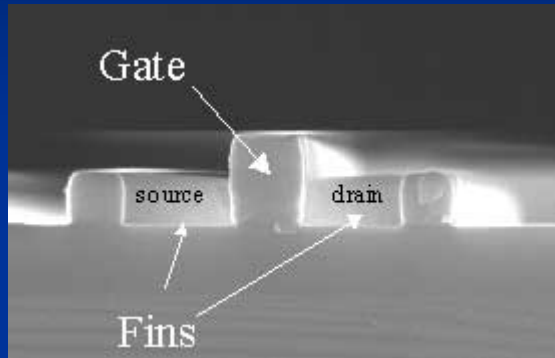


Principes du MOS

- Le transistor MOS se comporte comme un interrupteur « presque » parfait
- MOS : Metal Oxyde Semiconductor

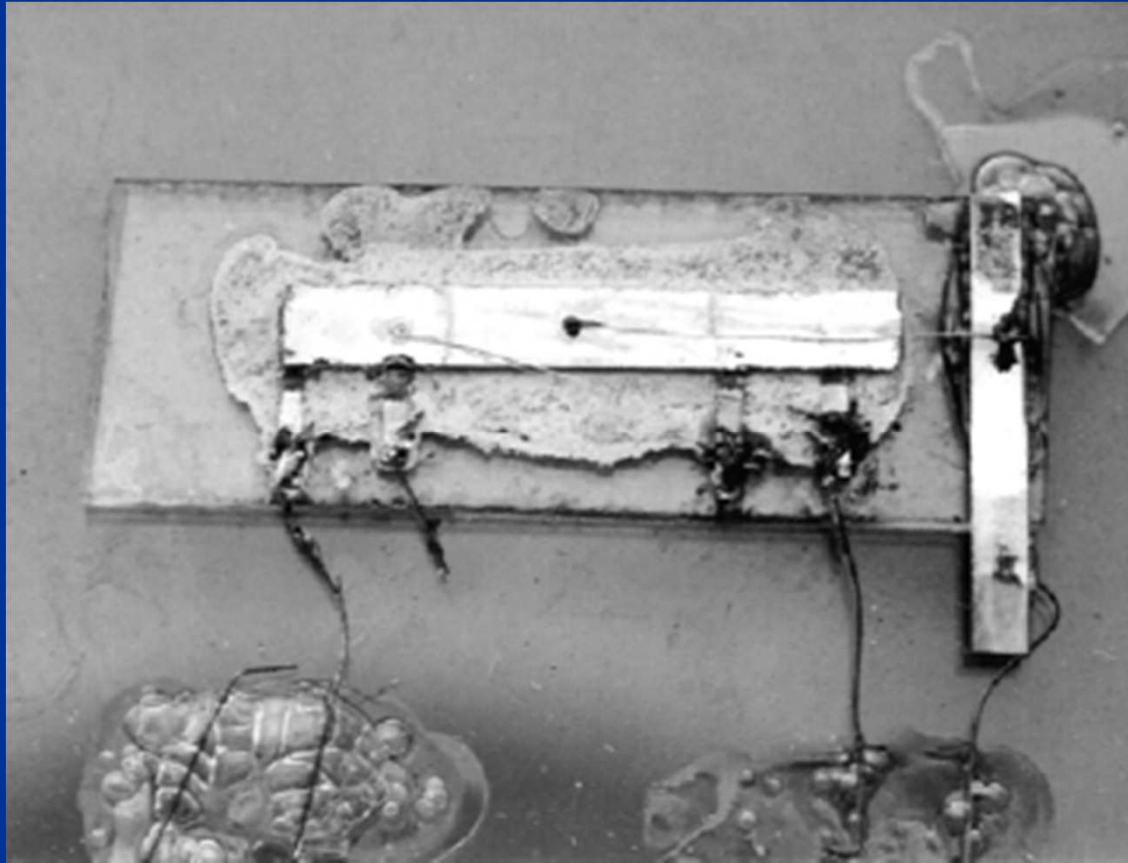


Coupe MOS

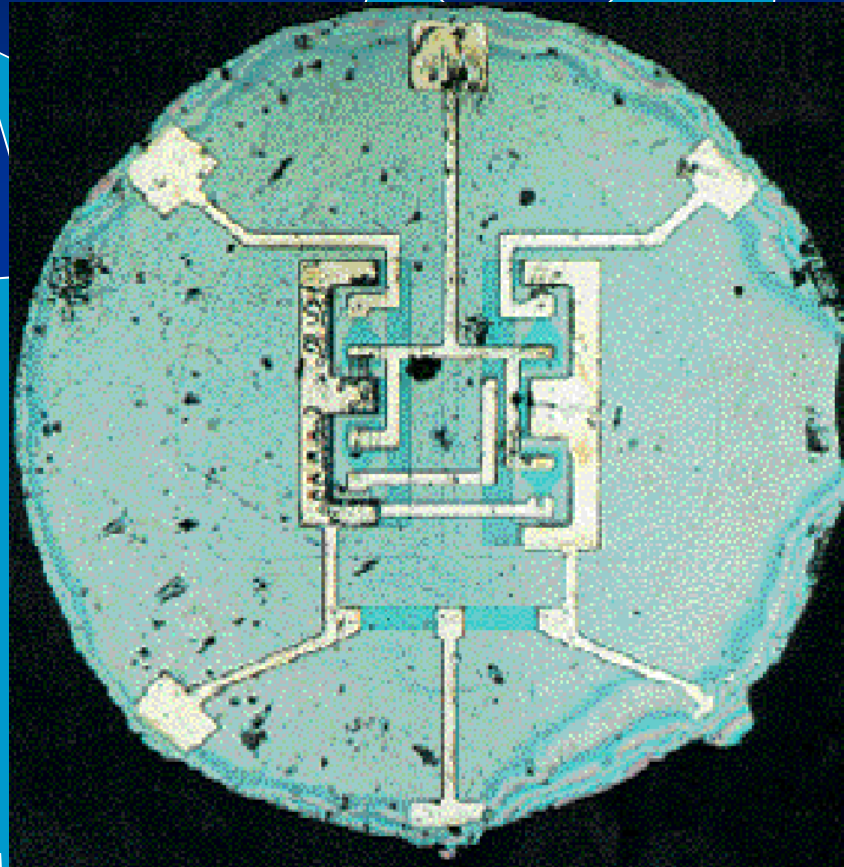


Premier Circuit Integre 1958

- Texas Instruments fabrique le premier CI :
- un inverseur à deux transistors

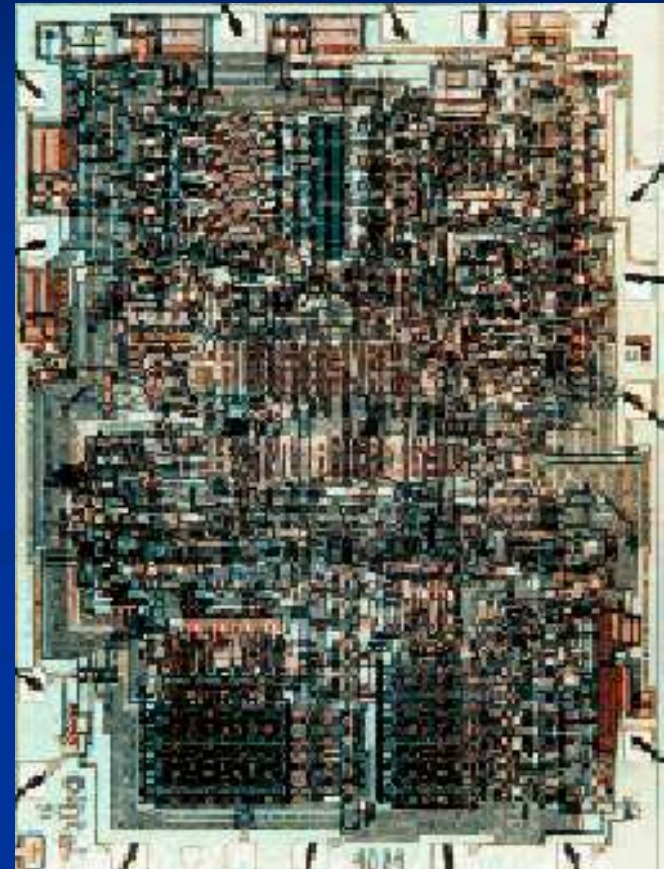
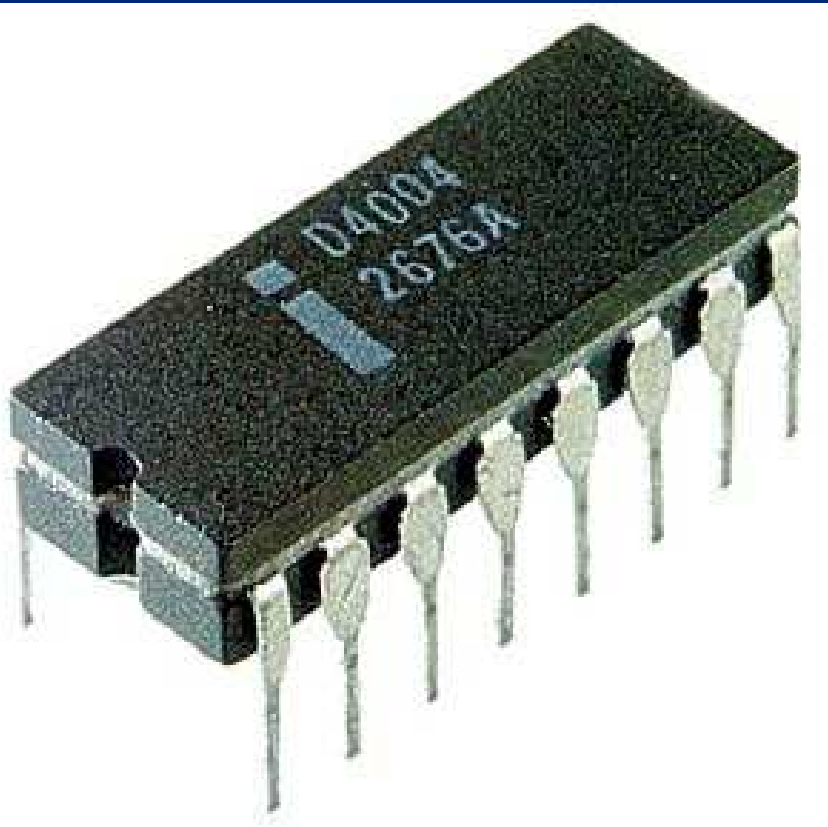


Explosion technologique : 1961



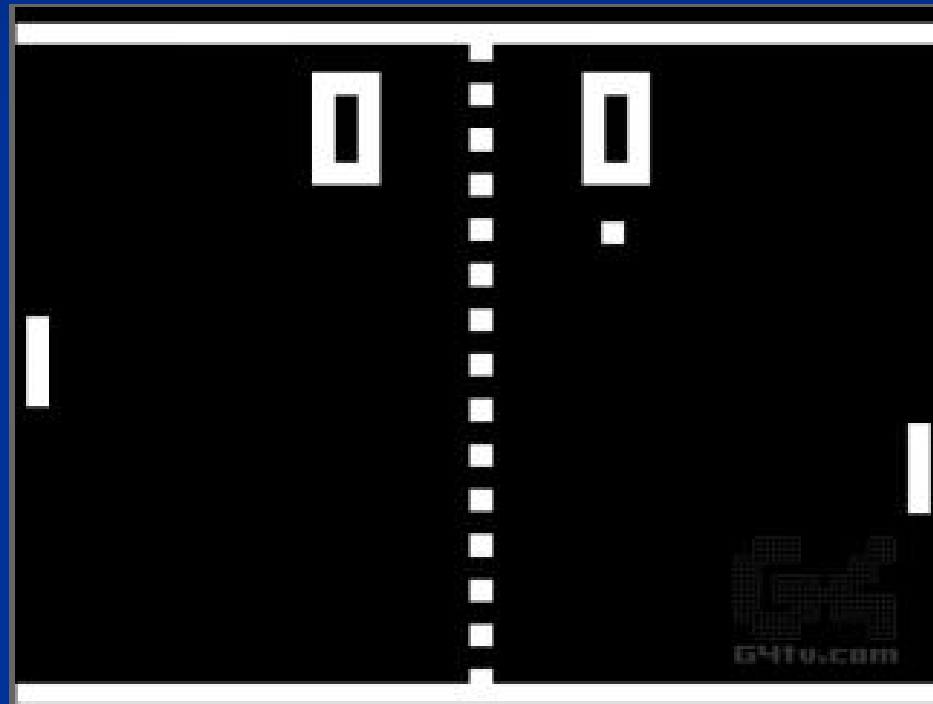
inverseur
4 transistors

INTEL 4004 le premier uProcesseur : 1971



15 novembre 1971
108KHZ **2300** Transistors (valeur actuelle \$20000)
technologie 10um (10000nm)

INTEL 8085 MOTOROLA 6809 : 1978



INTEL 8086 MOTOROLA 68000 : 1982



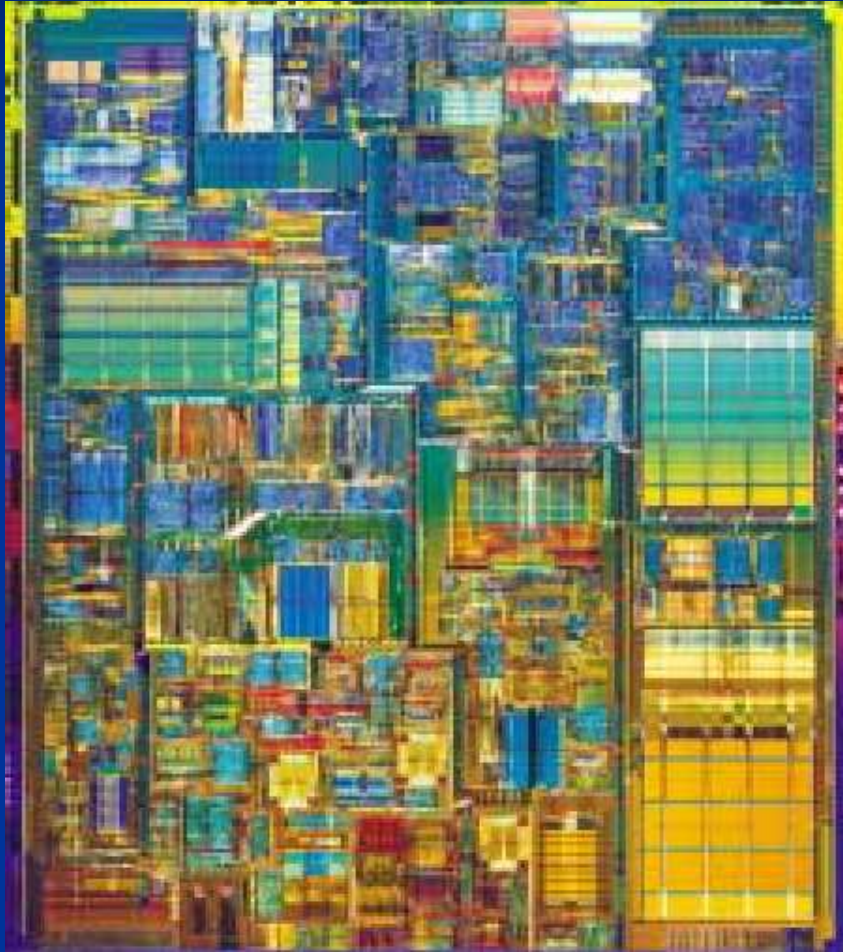
INTEL 80386 et 486 MOTOROLA 68020 : 1988



1996 PENTIUM POWERPC



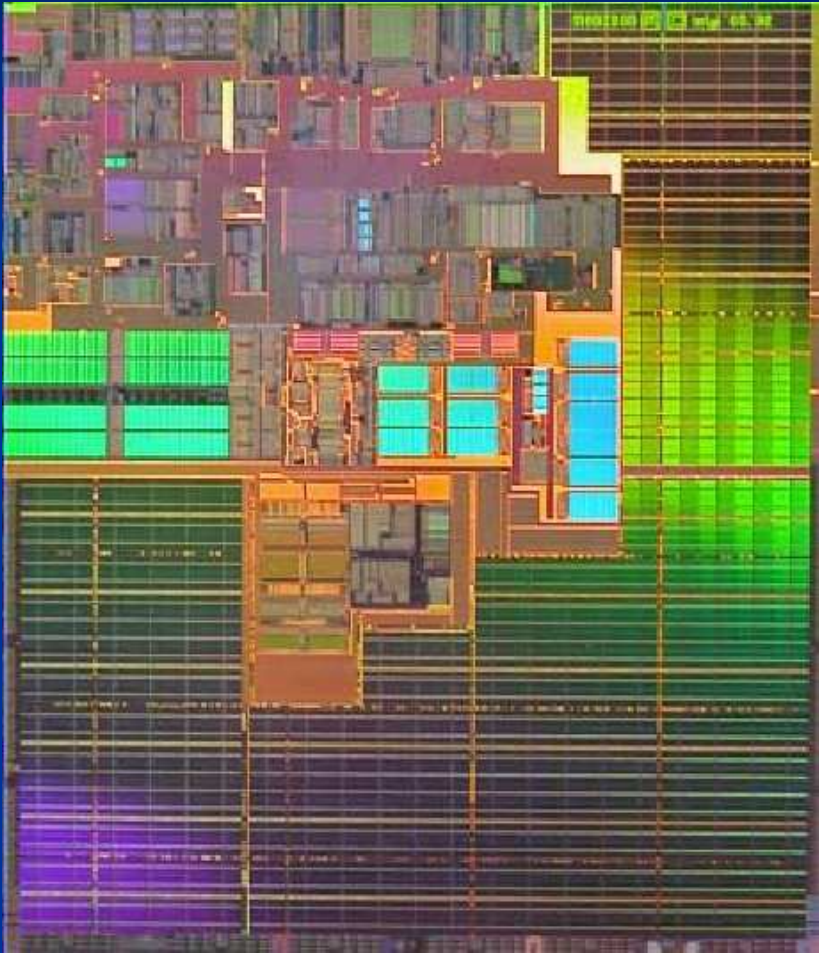
2002



2007 PENTIUM 4

44 Millions
de transistors

2008



Processeur INTEL
ITANIUM

Process 130nm

6MB de RAM CACHE

410 Millions de transistor

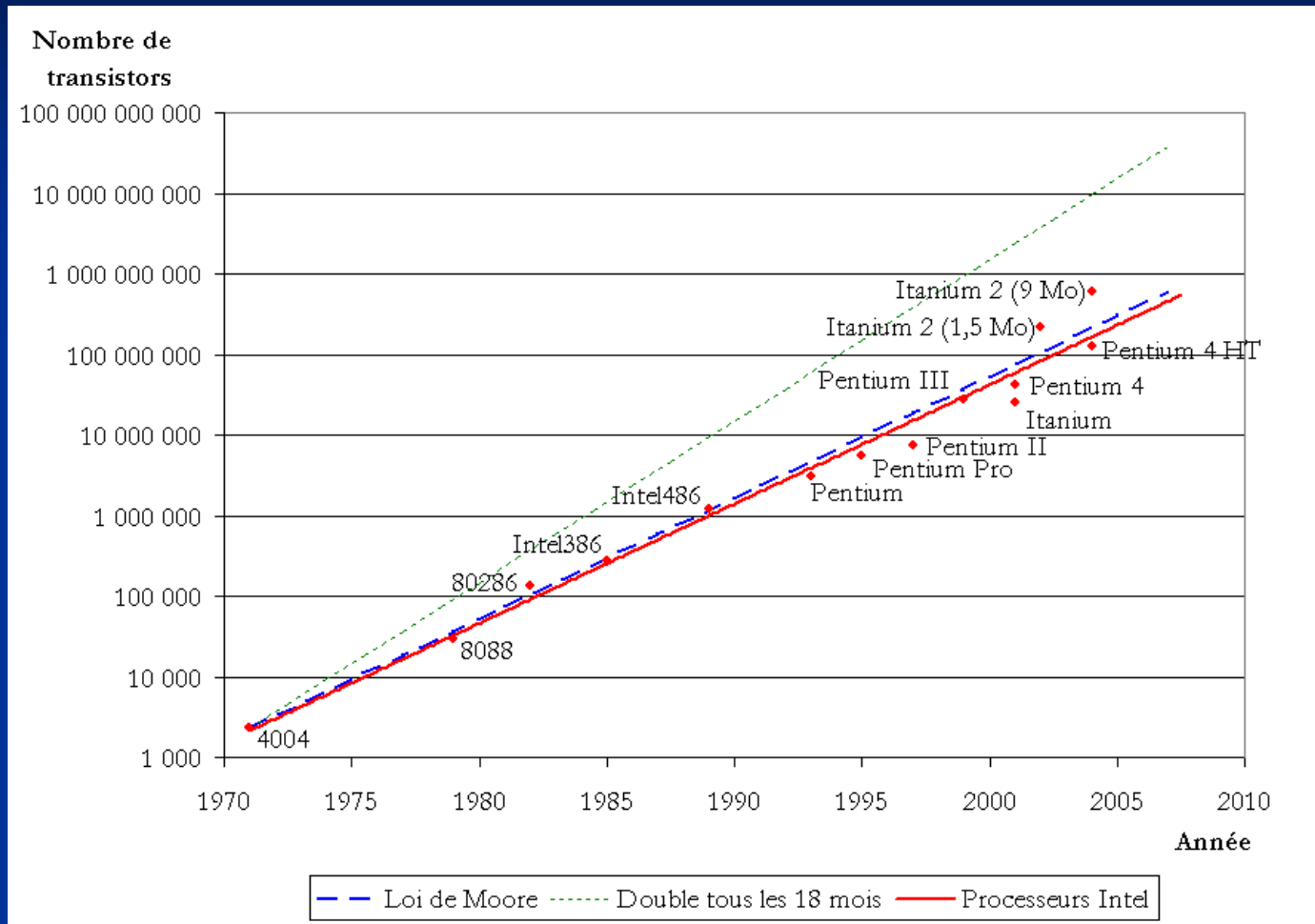
Cheveux humain 100 μm

Virus Amoeba 15 μm

Globule rouge 7 μm

Virus du SIDA 100 nm

La loi de Moore



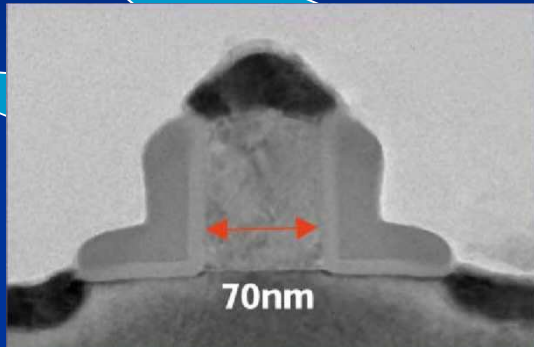
Le nombre de transistors intégrés sur une même surface double tous les 24 mois

Technologie

Caractéristiques	1999	2001	2004	2008
Process (nm) Drain-Source	180	130	90	60
Transistors par puce (Millions)	24	48	135	539
Fréquences de bus	1200	1600	2000	2700
Nombres de couches	9	7	8	9

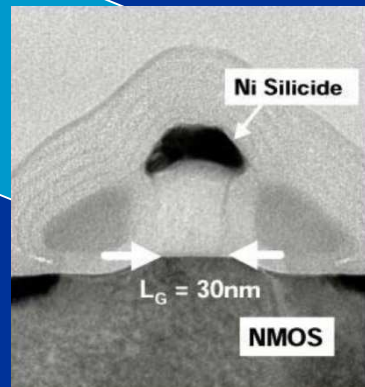
Evolution

70nm



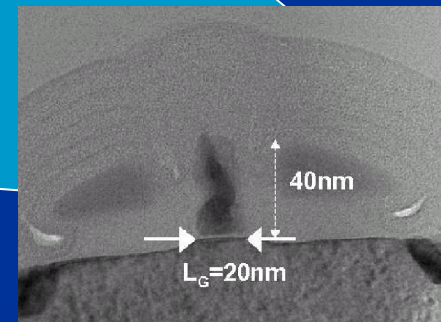
2004

30nm



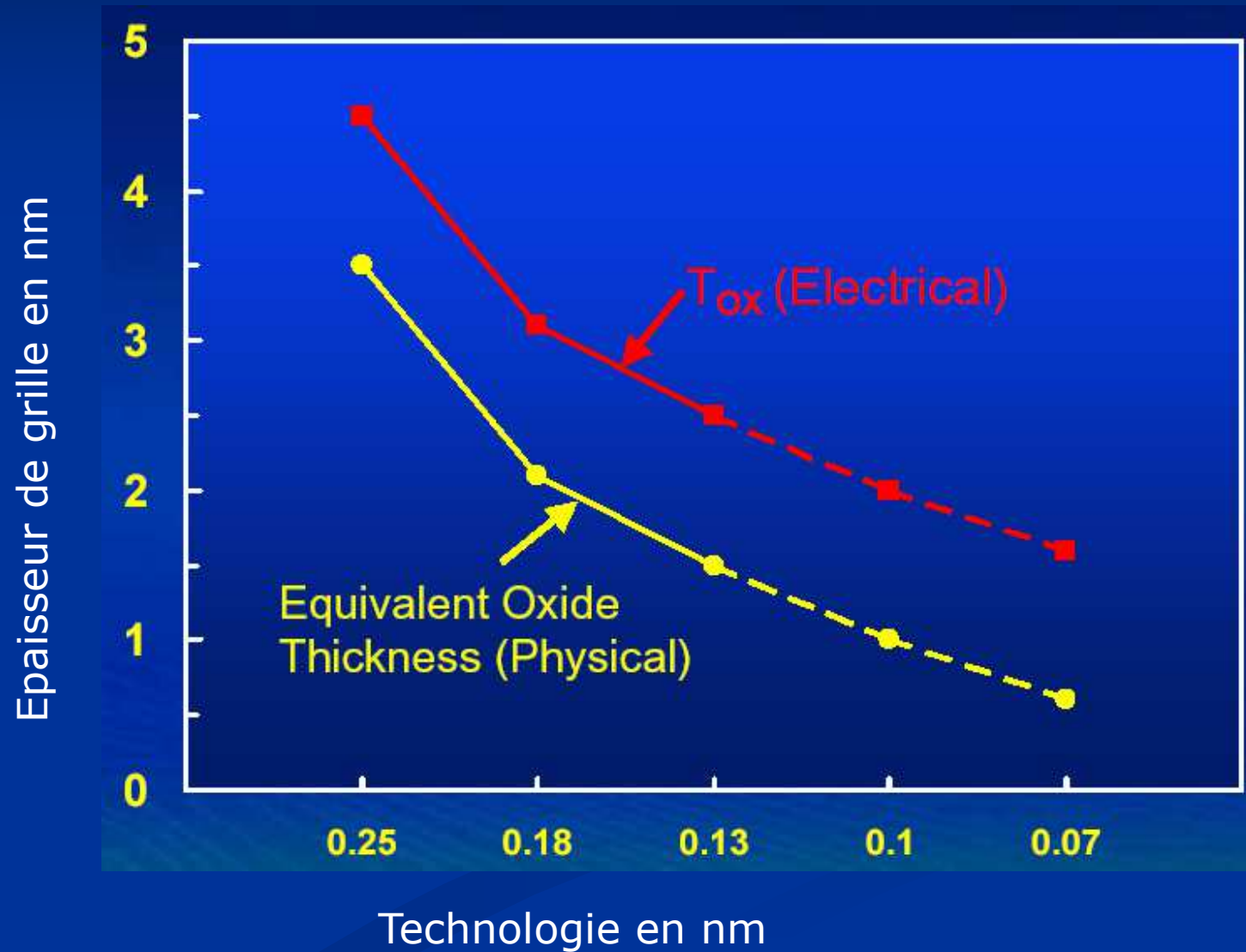
2007

20nm



2009

Epaisseur de grille / technologie



Epaisseur de grille

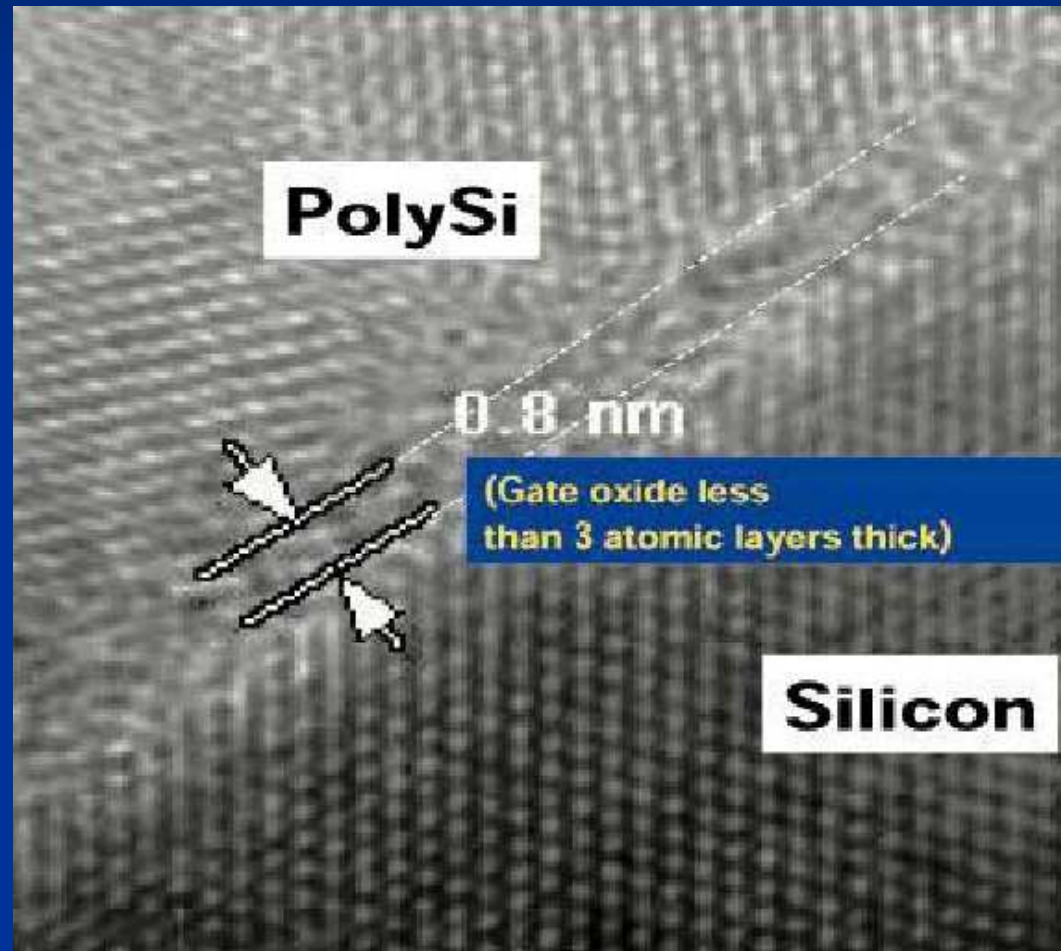
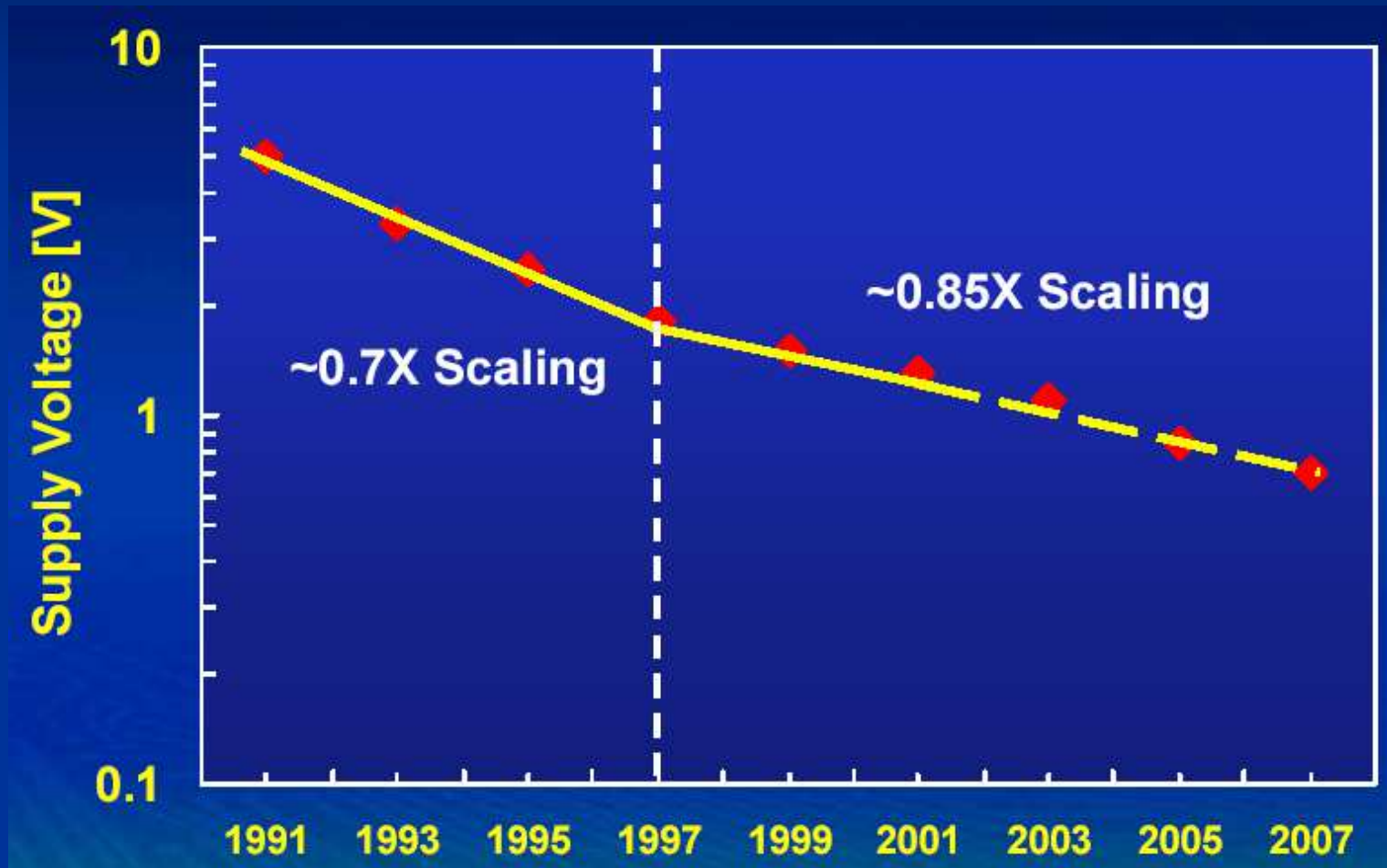
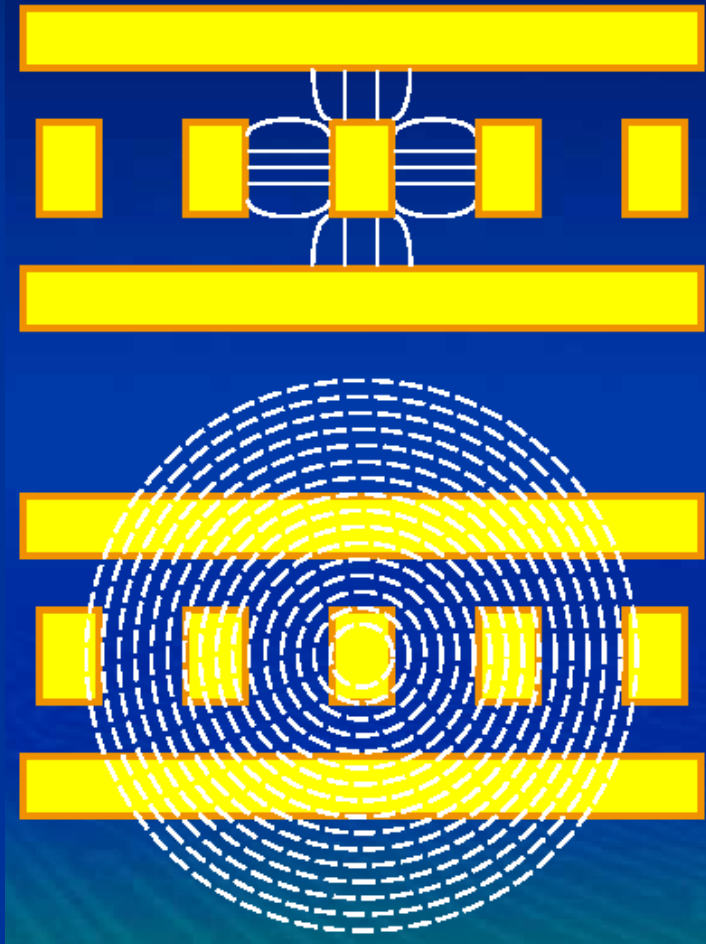


Image au microscope à effet tunnel

Tension d'alimentation



CEM



Couplage capacitif :

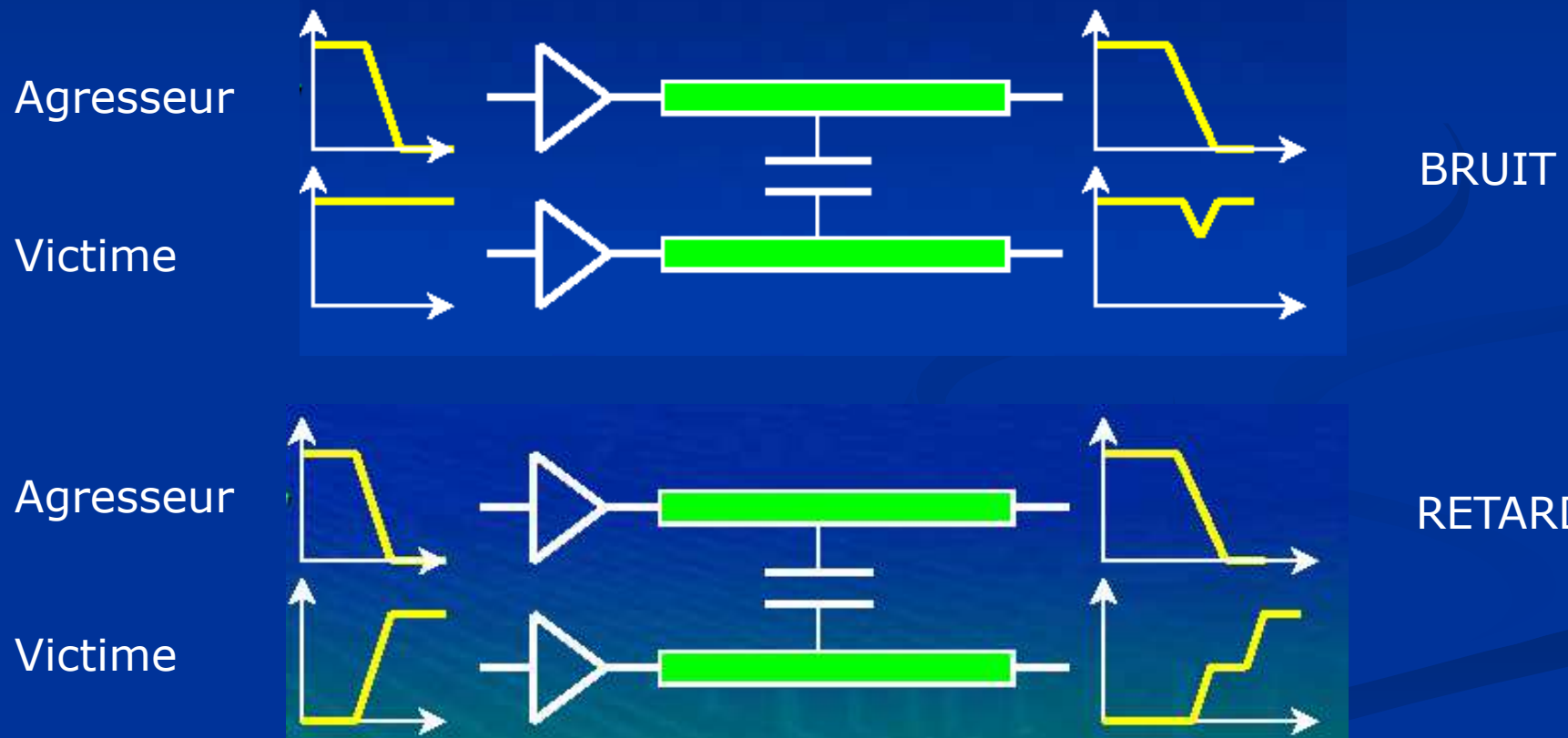
Du au champ électrique entre deux conducteur de potentiels différents

Couplage inductif :

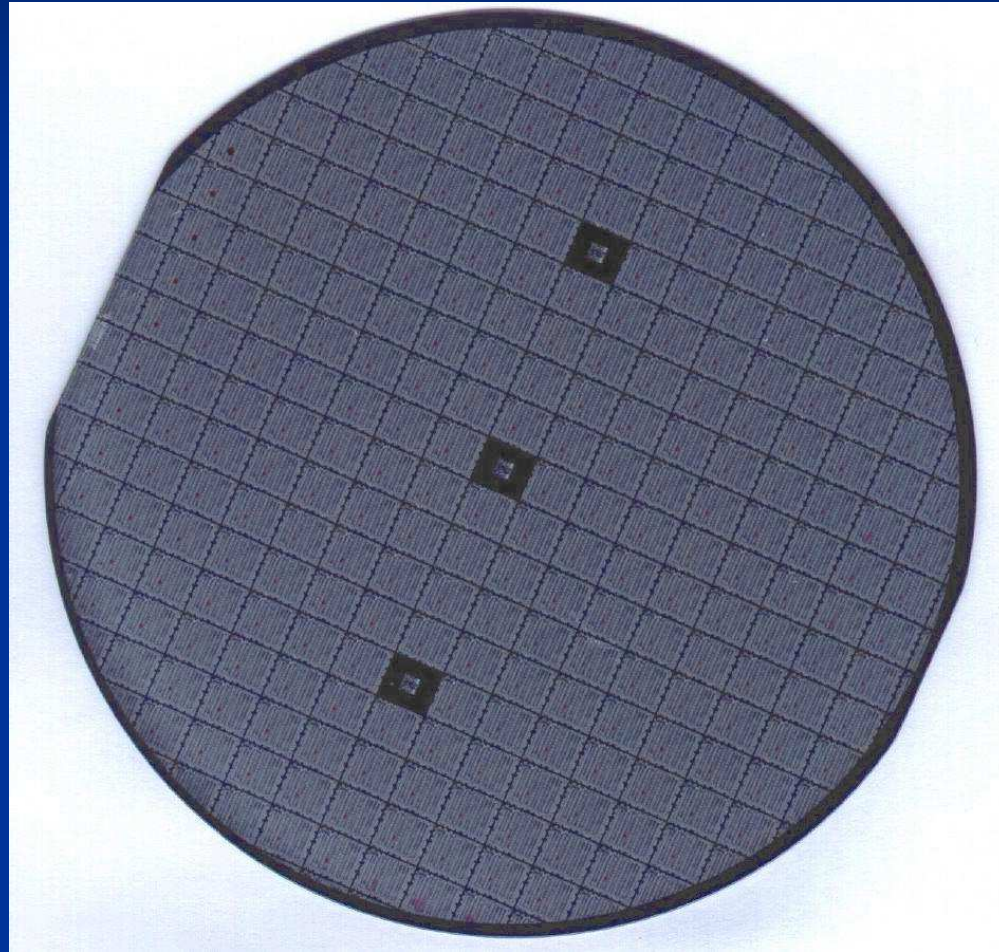
Du aux variations du champ magnétique lié aux variations de courant

Dépend de la fréquence

Les effets de la CEM



Fabrication de masse



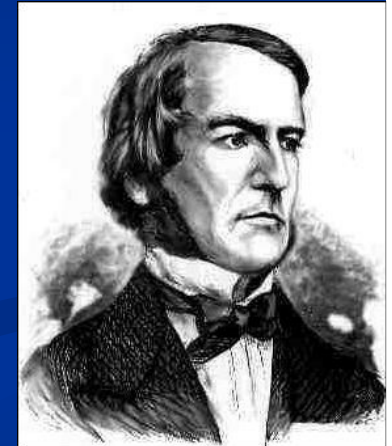
Coût des équipements



Lithographie par Ultra Violets

Technologie et calcul binaire

- L'évolution technologique a permis la mise en œuvre de l'algèbre de Georges Boole (1815-1864) à travers la logique combinatoire et séquentielle.



- Construction de machines d'états (2^{ème} guerre mondiale) puis du premier micro-processeur (INTEL)

Utilisation électronique du calcul binaire

- Codage électronique des nombres avec deux états
- Réalisation d'opérations complexes à partir d'opérateurs Booléens simples (NON, ET, OU) réalisés avec des transistors.

Ecrire des nombres avec des BIT l'octet :

Bit	7	6	5	4	3	2	1	0
Val	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Déc	128	64	32	16	8	4	2	1

$$10110110 = 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0$$
$$= 128 + 32 + 16 + 4 + 2 = 182$$

$$10110110 = B6$$

1011	0110
B	6

Notation hexadécimale :

\$B6 ou B6h ou 0xB6

Codage

■ Les nombres peuvent être écrits en :

- Décimal : 123
- Hexadécimal : 7Bh ou 0x7B
- Binaire : 01111011b
- BCD (binary coded decimal).
 - 23 en BCD : 0010 0011 (2 et 3)

Dec	Hexa
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

8 bits représentent un octet (byte)
16 bits représentent un mot (word)
32 bits représentent un mot long (long)

Pour représenter les nombres signés (positifs ou négatifs)
on utilise le bit de poids fort

ex : non signé un octet est compris entre 0 et 255

signé il est compris en -128 et +127

Binaire signé	Dec
00000011	3
00000010	2
00000001	1
00000000	0
11111111	-1
11111110	-2
11111101	-3
11111100	-4
111110101	-5

ASCII

- *L'American Standard Code for Information Interchange* permet l'échange de caractères alphanumériques (lettres, chiffres, ponctuation, contrôles)

code	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0x00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	NP	CR	SO	SI
0x10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0x20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0x30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0x60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Attention, le chiffre '5' codé en ASCII vaut 35h, ce qui n'a rien à voir avec la valeur 5

Faire des OPERATIONS

EX : ADDITION BINAIRE

$$\begin{array}{r} \\ 1011 \\ + 1110 \\ \hline 11001 \end{array}$$

-X : rendre un nombre négatif
Le complément à 2

5 = 00000101

Inversion ! 11111010

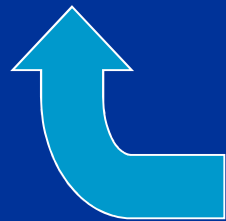
+1 -5 = 11111011

Inversion ! 00000100 +1 00000101



Un exemple : $-3+5$

$$\begin{array}{r} 1111101 \text{ (-3)} \\ + 0000101 \text{ (5)} \\ \hline 10000010 \text{ (2)} \end{array}$$



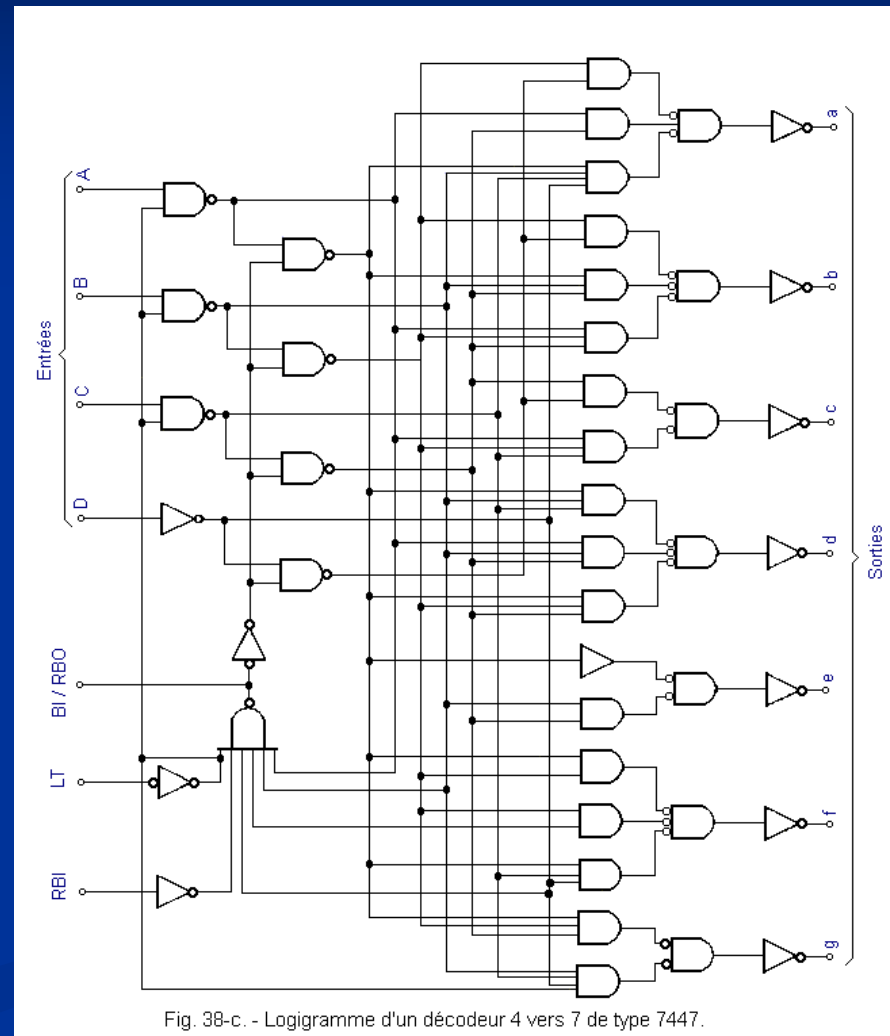
RETENUE

Fonction logiques

Les opérations sont réalisables à partir de fonctions logiques



OU OUI NON
OU-EXCLUSIF



AND : &

Bit1	Bit2	AND
0	0	0
0	1	0
1	0	0
1	1	1

Masquage

PORTA	x	x	x	x	x	x	x	x
&	0	0	0	1	0	0	0	0
=	0	0	0	x	0	0	0	0

Forçage à 0

PORTA	x	x	x	x	x	x	x	x
&	1	1	1	0	1	1	1	1
=	x	x	x	0	x	x	x	x

OR : |

Bit1	Bit2	OR
0	0	0
0	1	1
1	0	1
1	1	1

Forçage à 1

PORTA	x	x	x	x	x	x	x	x
OU	0	0	0	1	0	0	0	0
=	x	x	x	1	x	x	x	x

XOR : ^

Bit1	Bit2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

BASCULEMENT

PORTA	x	x	x	x	x	x	x	x
XOR	0	0	0	1	0	0	0	0
=	x	x	x	/x	x	x	x	x

DECALAGE : <<

0 1 0 1 0 0 1 1

 ← 0 ← 1 ← 0 ← 1 ← 0 ← 0 ← 1 ← 1 ← 0

1 0 1 0 0 1 1 0

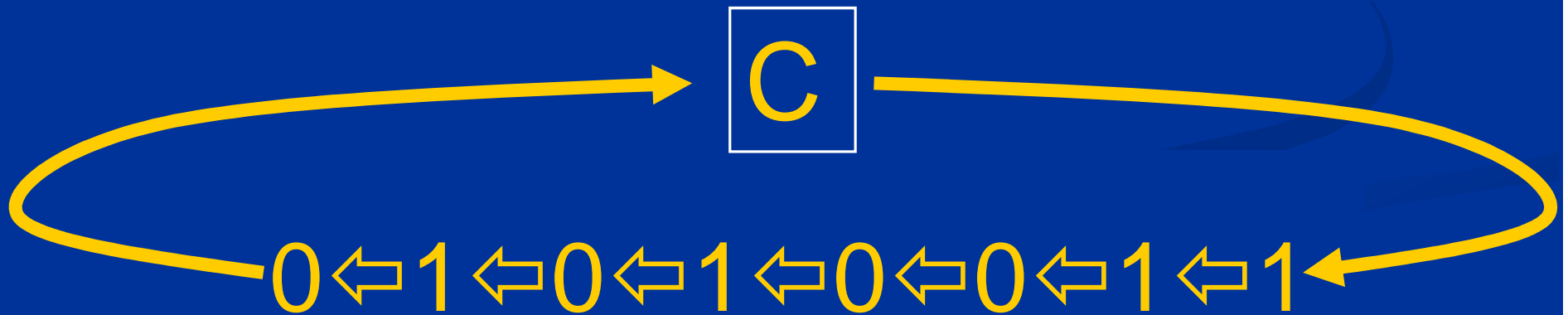
ROTATION

0 1 0 1 0 0 1 1

C

0 ← 1 ← 0 ← 1 ← 0 ← 0 ← 1 ← 1

1 0 1 0 0 1 1 C



QUIZZ



5Ah + 78h = () hexa

10011010
-01111000

 Interpréter le résultat suivant que les nombres soient signés ou non ?

10110001
+ 10001101

 Y a t-il overflow ? comment y palier ?

Combien de bits pour représenter le nombre 1492 ?

Un double mot est constitué de combien de bits ?

Combien de koctets représente une mémoire de 256 kbits ?

Combien de cases mémoires possède une mémoire de 64 koctets ?

Intérêt du code hexadécimal ?

Dans une machine à laver, quel composant est le mieux adapté: un microprocesseur, un microcontrôleur ou un DSP ?

Dans un analyseur de spectre, quel composant est le mieux adapté ?

Pour commander un moteur, quel composant est le mieux adapté ?

Quels sont les avantages des microcontrôleurs

Donnez un exemple d'utilisation de la fonction XOR

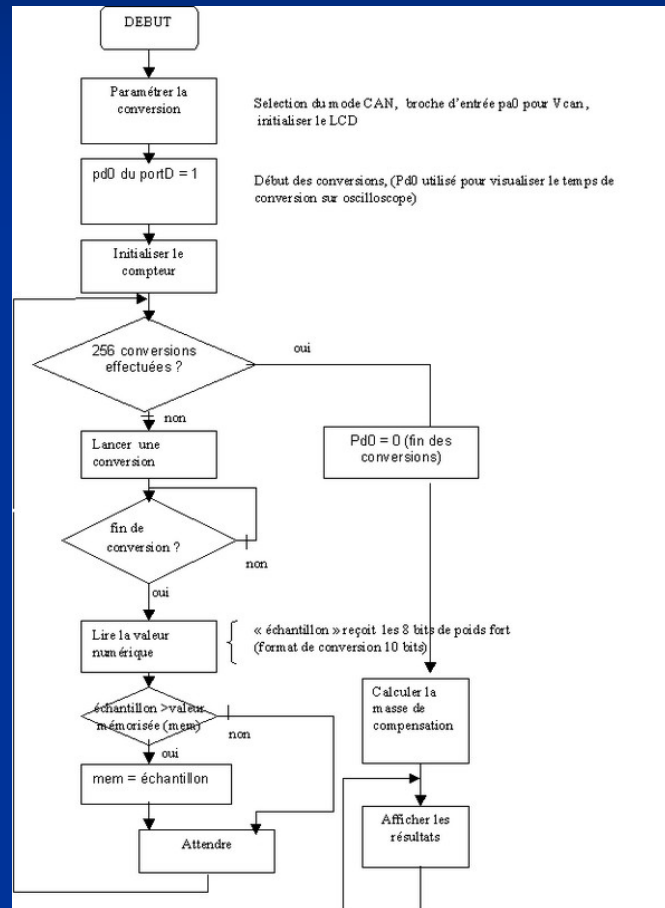
Quel est l'ensemble des nombres entiers signés codés sur 16bits

QUIZZ

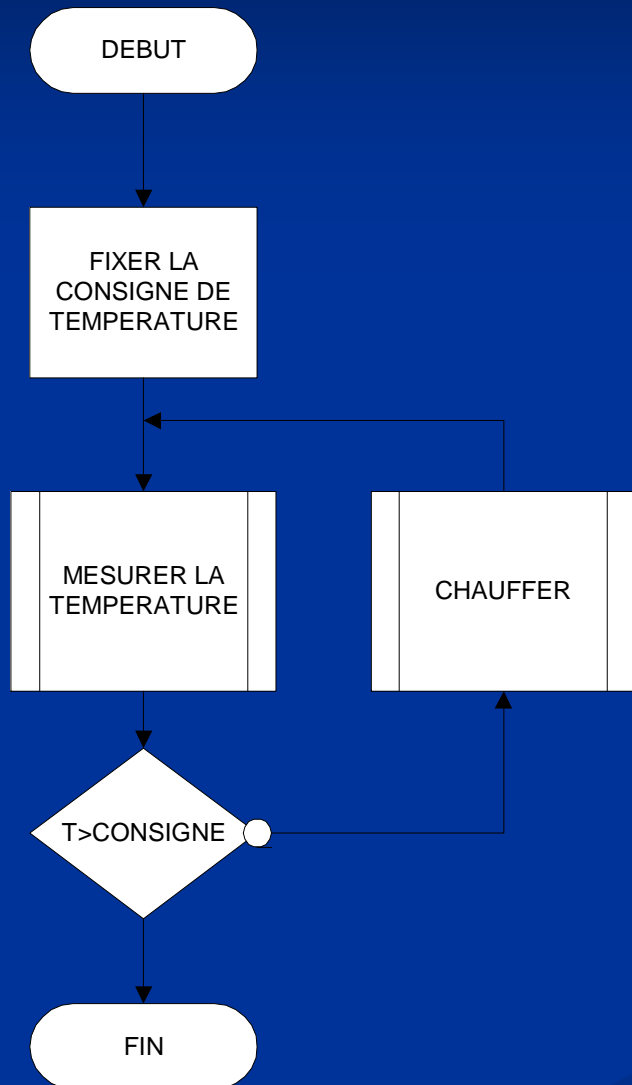
- Qu 'appelle-t-on un système embarqué ?
- Qu 'appelle-t-on un microcontrôleur 8 bits ?
- Qu 'appelle-t-on un périphérique ?
- Pourquoi les fréquences d 'horloge des microcontrôleurs sont-elles relativement faible ?

- Quelle est l'ordre de grandeur de l'horloge de cadencement d'un microcontrôleur ?
- Comment choisit-on un microcontrôleur ?
- Quelles différences faites vous entre un microprocesseur et un microcontrôleur ?

Éléments d'algorithmie



Algorithme / Algorithm



Debut

Fixer la température de consigne

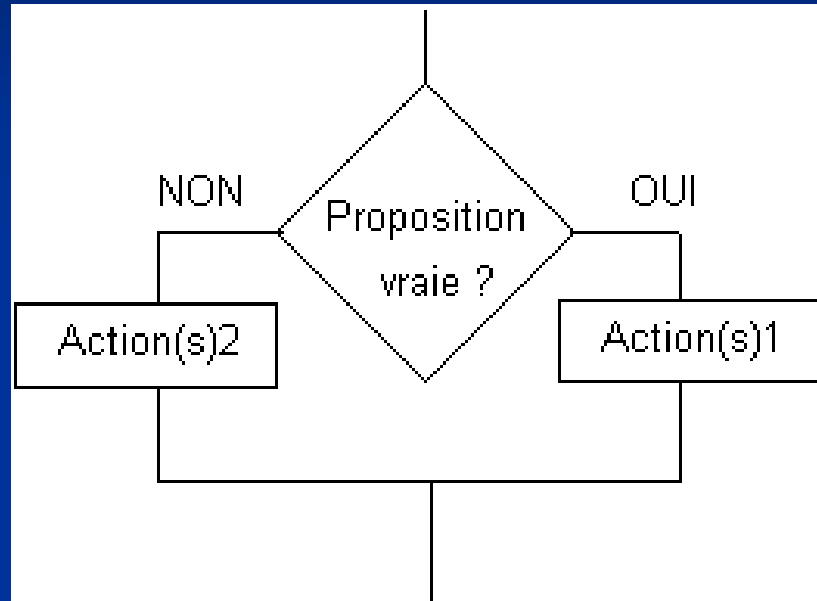
Tant que $T < \text{Consigne}$

Chauffer

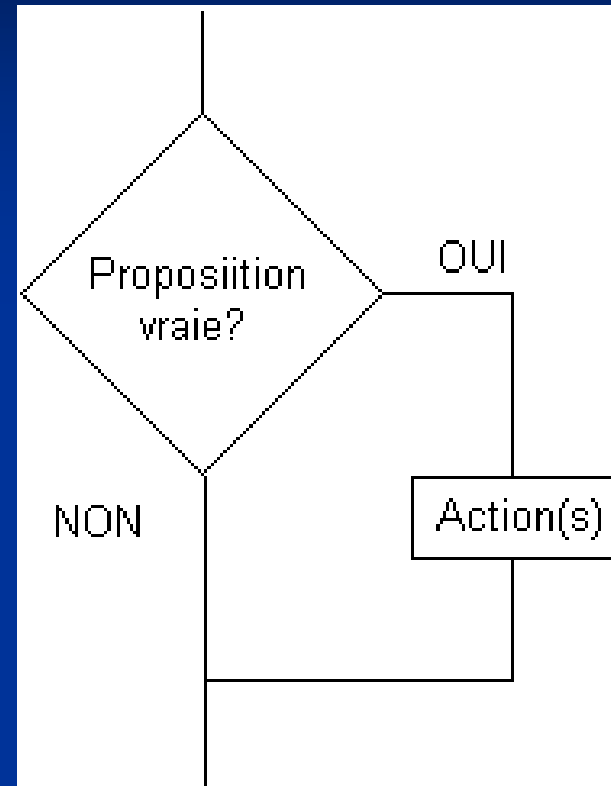
Fin Tant que

Fin

Structures alternatives

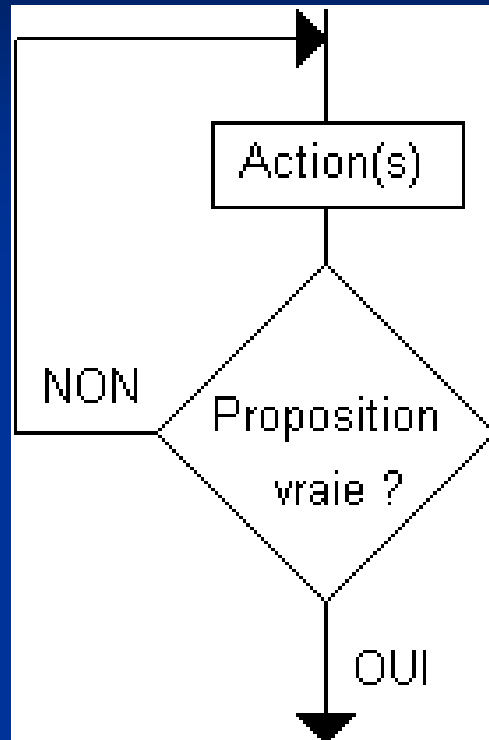


SI *proposition vraie*
alors *action(s)1*
sinon *action(s)2*
Fin si

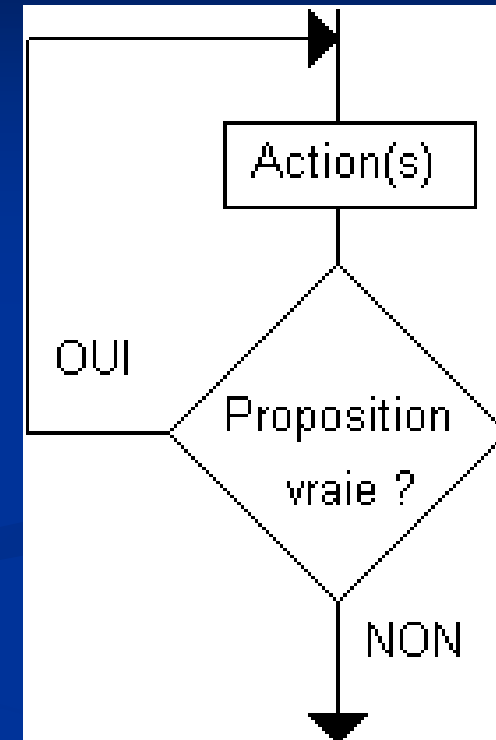


SI *proposition vraie*
alors *action(s)*
Fin si

Structures répétitives



Répéter *action(s)*
jusqu'à *proposition vraie*



Répéter *action(s)*
tant que *proposition vraie*

Langages

C

```
char heure,min,sec;  
int pression,temp;  
char* calcul(int m)  
{ char c;  
  m=valeur/1000;  
  c=(valeur-(1000*m))/100;  
  if (c>10) return (c );  
}
```

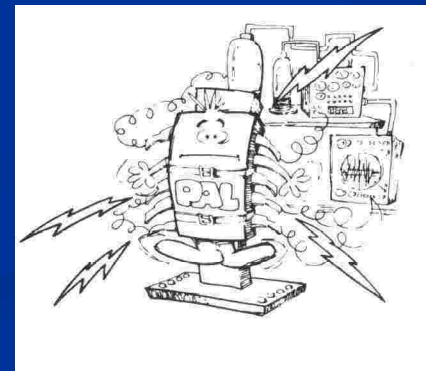


ASSEMBLEUR

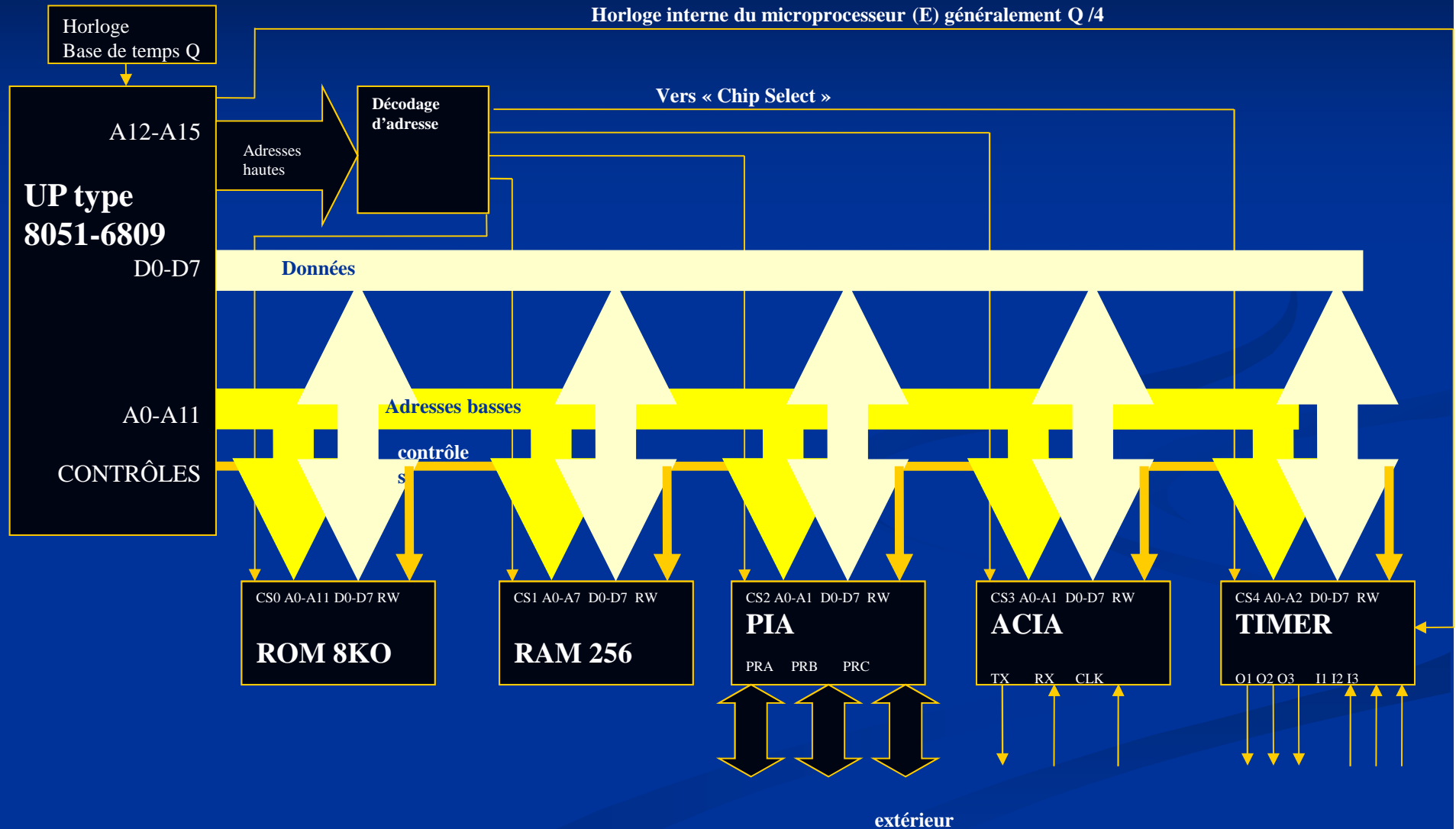
```
LED      EQU      P5.0  
CSEG     AT 0  
         MOV SP ,#7Fh      ;  
  
SUITE:   CPL LED  
CALL    TEMPO  
SJMP    SUITE  
  
TEMPO:   MOV R0 ,#0FFh  
  
TEMPO1:  MOV      R1 ,#0FFh  
TEMPO2:  DJNZ     R1 ,TEMPO2  
         DJNZ     R0 ,TEMPO1  
         RET
```

Langage machine:

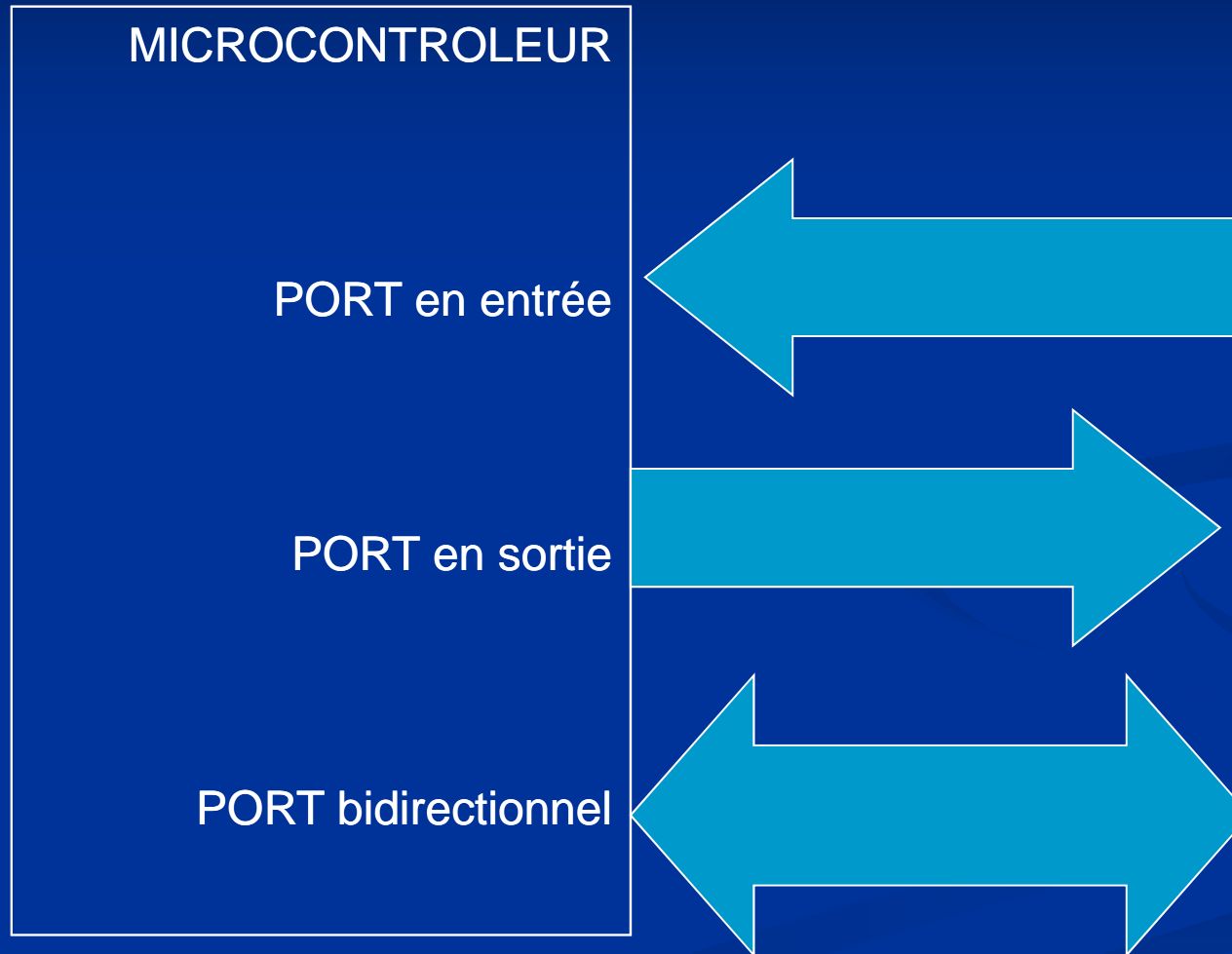
0x0000	75817F
0x0003	B2A0
0x0005	1109
0x0007	80FA
0x0009	78FF
0x000B	79FF
0x000D	D9FE
0x000F	D8FA
0x0011	22



Systeme minimum



PORTS PARALLELES



Lexique

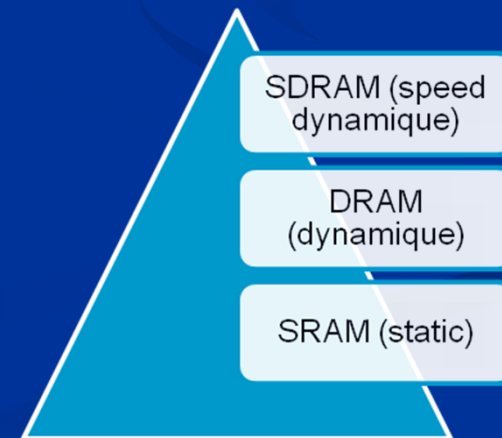
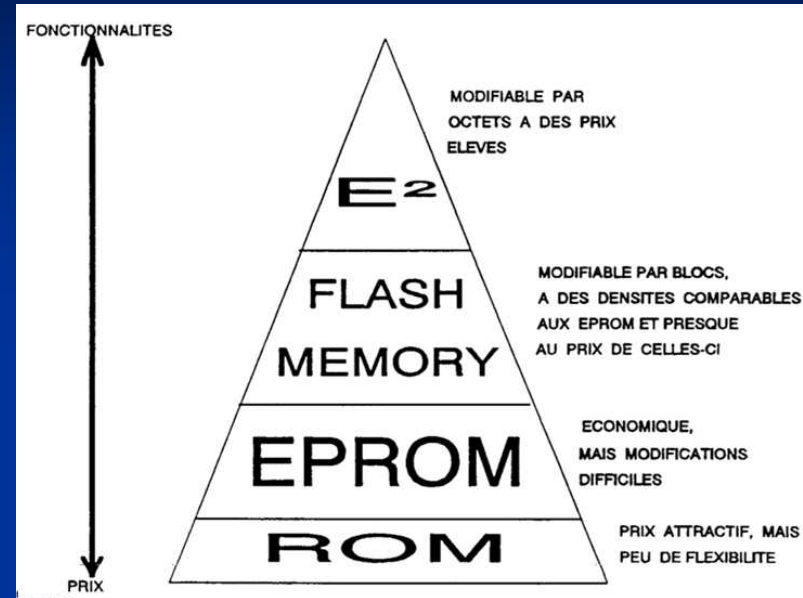
- **UP:** 8 bits, 64KO type 8051, 68HC11, PIC16, PIC18, ATMEL AVR , ST6, ST7...
- **ROM:** mémoire morte PROM ou EPROM, elle contient le programme
- **RAM :** Mémoire vive statique , elle contient les données (variables)
- **PIA, PIO :** Peripheral interface adapter, périphérique input/output. Port parallèle 8bits
- **UART , USART :** Asynchronous communications interface adapter ou Universal Synchrone/Asynchrone Receive/Transiver. Port série code NRZ .
- **TIMER :** PTM : Programmable Timer Module . Production et mesures de durées.
- **Décodeur d'adresses :** permet à partir du bus d'adresse (A0-A15) ou d'une partie de celui ci de créer des signaux permettant de sélectionner les différents boîtiers (CS pour Chip Select) lorsque l'adresse du boîtier considéré est active.

Interfaces Internes/Externes

- EEPROM : mémoire morte effaçable électriquement
- ADC : convertisseurs analogiques numériques
- DAC : convertisseurs numériques analogiques
- Comparateurs de tension
- Horloge temps réel (RTC)
- Port parallèle de puissance
- Interface série synchrone SPI ou IIC
- Gestionnaire afficheur LCD graphique / alphanumérique
- Interface ethernet
- Interface WIFI
- Bus de communications de terrain (CAN ou LIN)
- Superviseur d'alimentation / Chien de garde (watch dog)

Mémoires

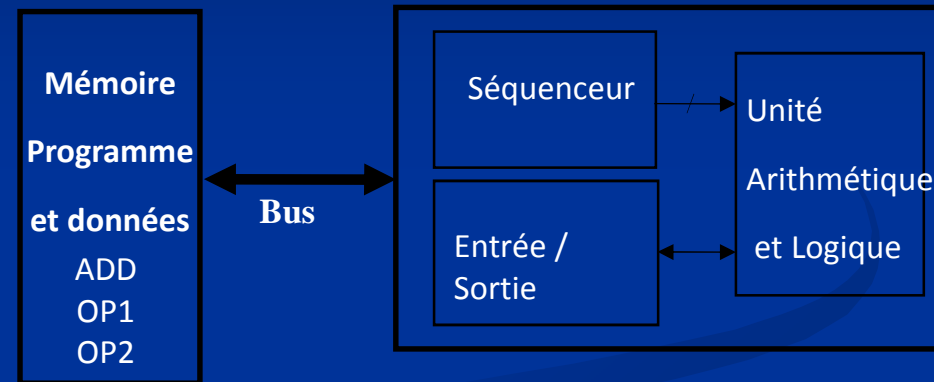
- Read Only Memory (ROM), elle conserve ses données sans énergie.
- Random Access Memory (RAM), elle ne conserve pas ses données sans énergie



Technologie des microcontrôleurs

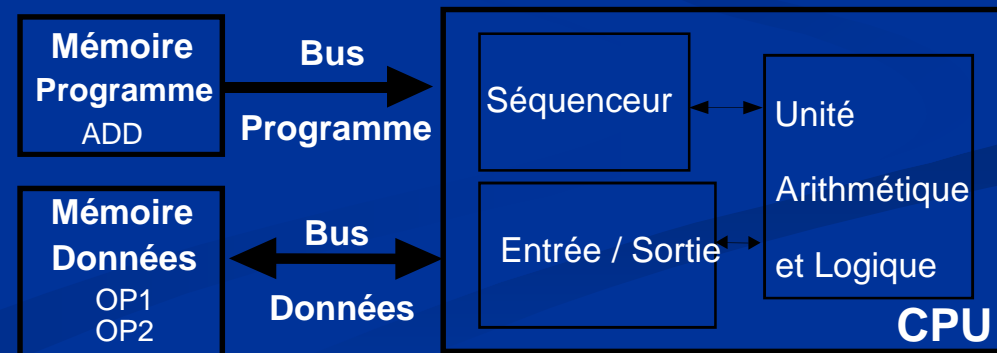
■ Von Neumann

John von Neumann (1903-1957),
mathématicien et physicien américain
d'origine hongroise



■ Harvard

université américaine située
à Cambridge au Massachusetts.



Deux concepts de jeu d'instructions pour les microcontrôleurs

- **CISC** (Complex Instruction Set Computer), 8051- 68HC11-ST6...

- ⇒ Compilateur simple

- ⇒ Architecture matérielle simple

- ⇒ Jeux d'instructions riches

- ⇒ Au programmeur d'optimiser le code

- **RISC** (Reduced Instruction Set Computer), PIC16, PIC18, ATMEL AVR, tous les DSP

- ⇒ Compilateur complexe

- ⇒ Architecture matérielle optimisées

- ⇒ Jeux d'instructions succincts

- ⇒ C'est le compilateur qui optimise

MICROCONTROLEURS – un système fermé

■ Avantages

Intégration dans un seul boîtier

Fiabilité

Coût de câblage réduit

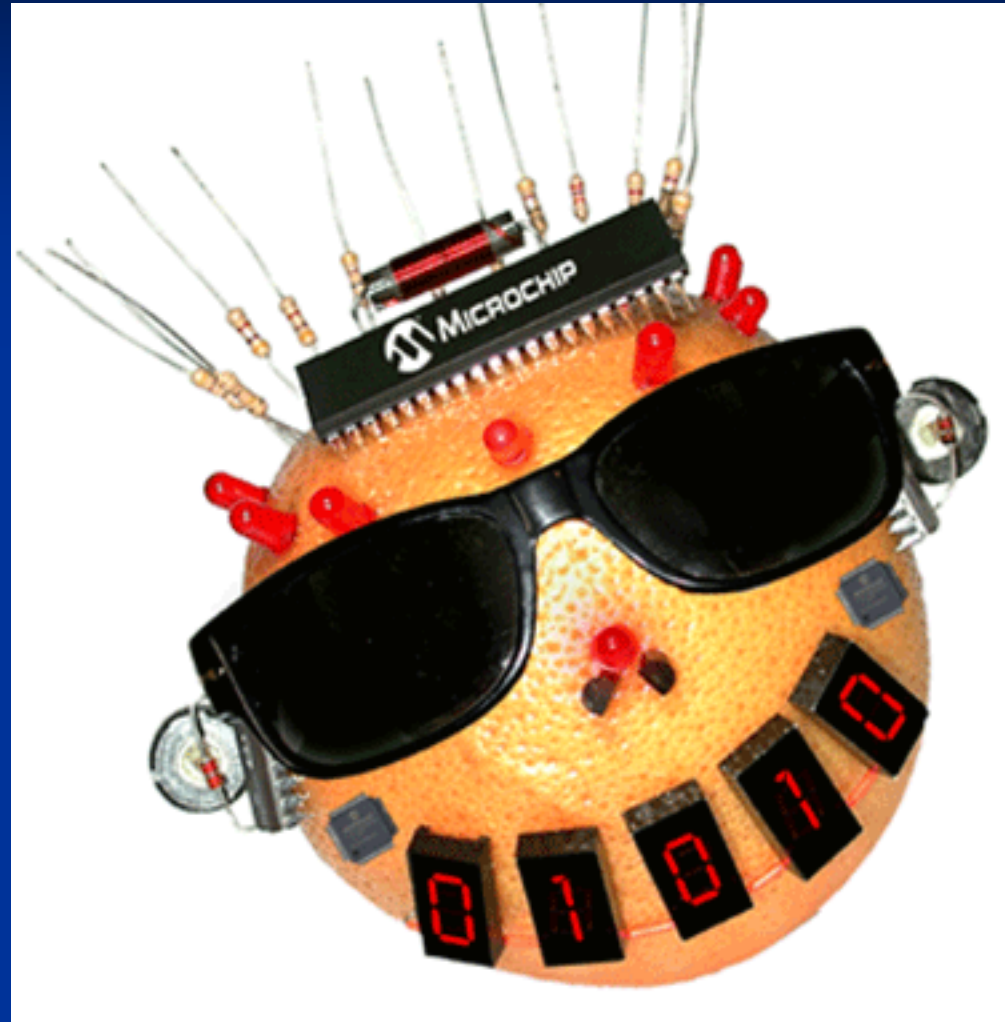
Faible consommation

■ Inconvénients

L'intégration de nombreux périphériques, de RAM, de ROM limite la puissance de calcul et la vitesse ces circuits.

Mise en œuvre et approche du composant d'apparence complexe.

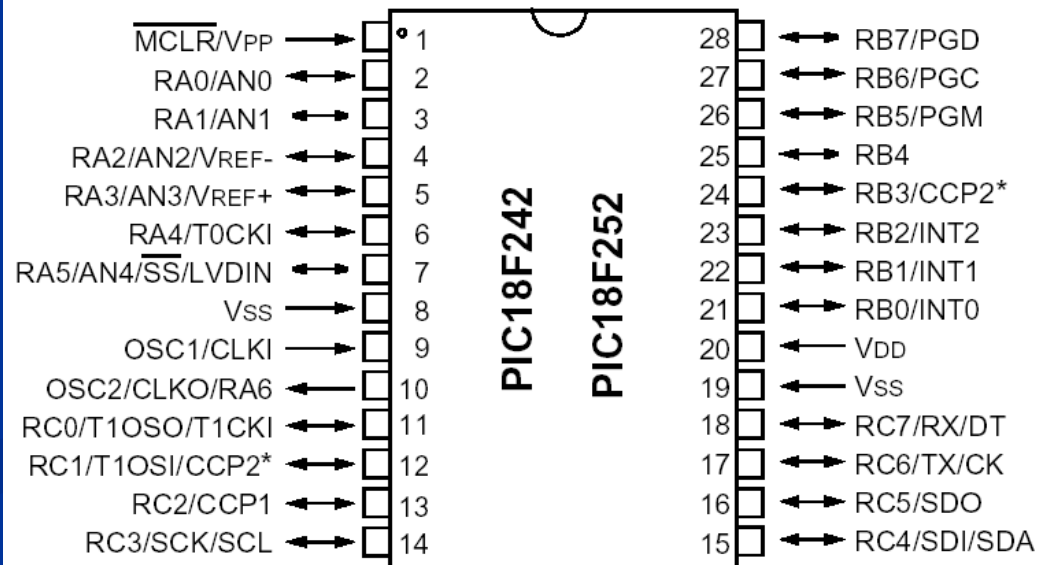
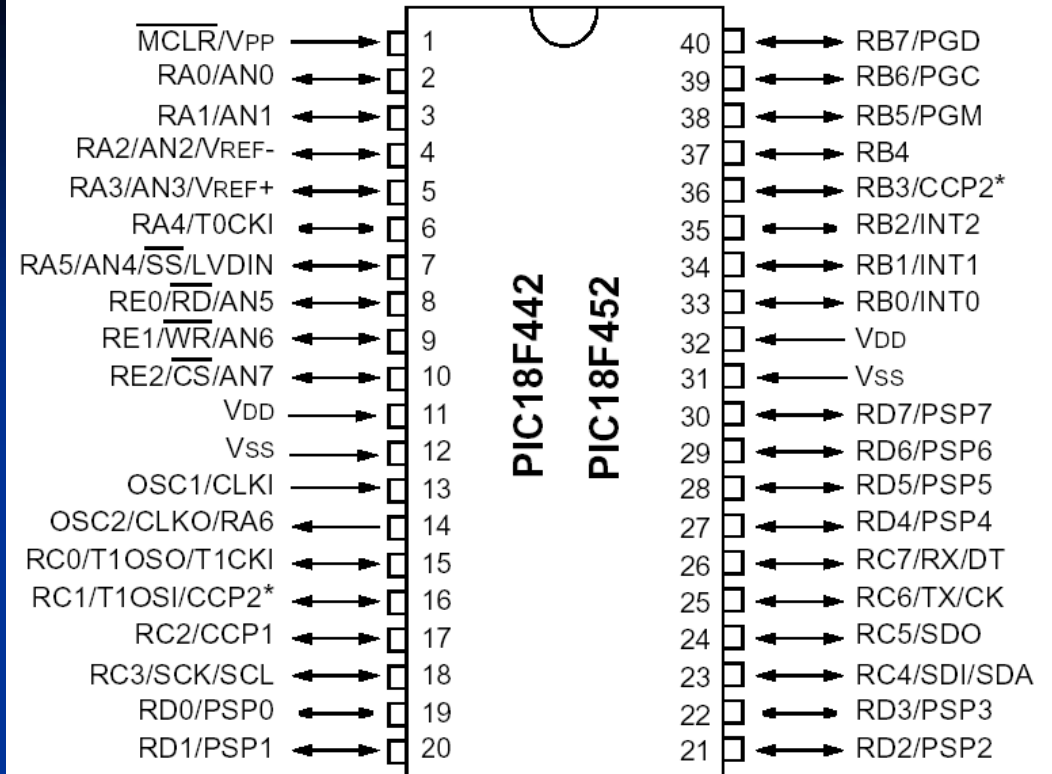
Les PIC18Fxxx généralités



CONSTITUTION

Caractéristiques	PIC18F2xxx	PIC18F4xxx
Fréquence Horloge MHz	DC-40 MHz	DC-40 MHz
Mémoire programme FLASH	64KO	64KO
Programme (Instructions)	32768	32768
Mémoire données	3KO	3KO
Mémoire EEPROM	1024 Octets	1024 Octets
Interruptions	17	18
Ports parallèles	A,B,C	A,B,C,D,E
Timers	4	4
Capture/Compare/PWM	2	2
Communications séries	SPI / I2C / USART	SPI / I2C / USART
Communications Parallèles	—	PSP
CAN 10-bit	5 entrées	8 entrées
RESETS	POR, BOR,RESET Instruction,Stack Full,Stack Underflow (PWRT, OST)	POR, BOR,RESET Instruction,Stack Full,Stack Underflow (PWRT, OST)
Détection de VDD faible programmable	oui	oui
Instructions	75	75
Boitiers	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin PLCC 44-pin TQFP

Brochages



Architecture

ROM : mémoire programme

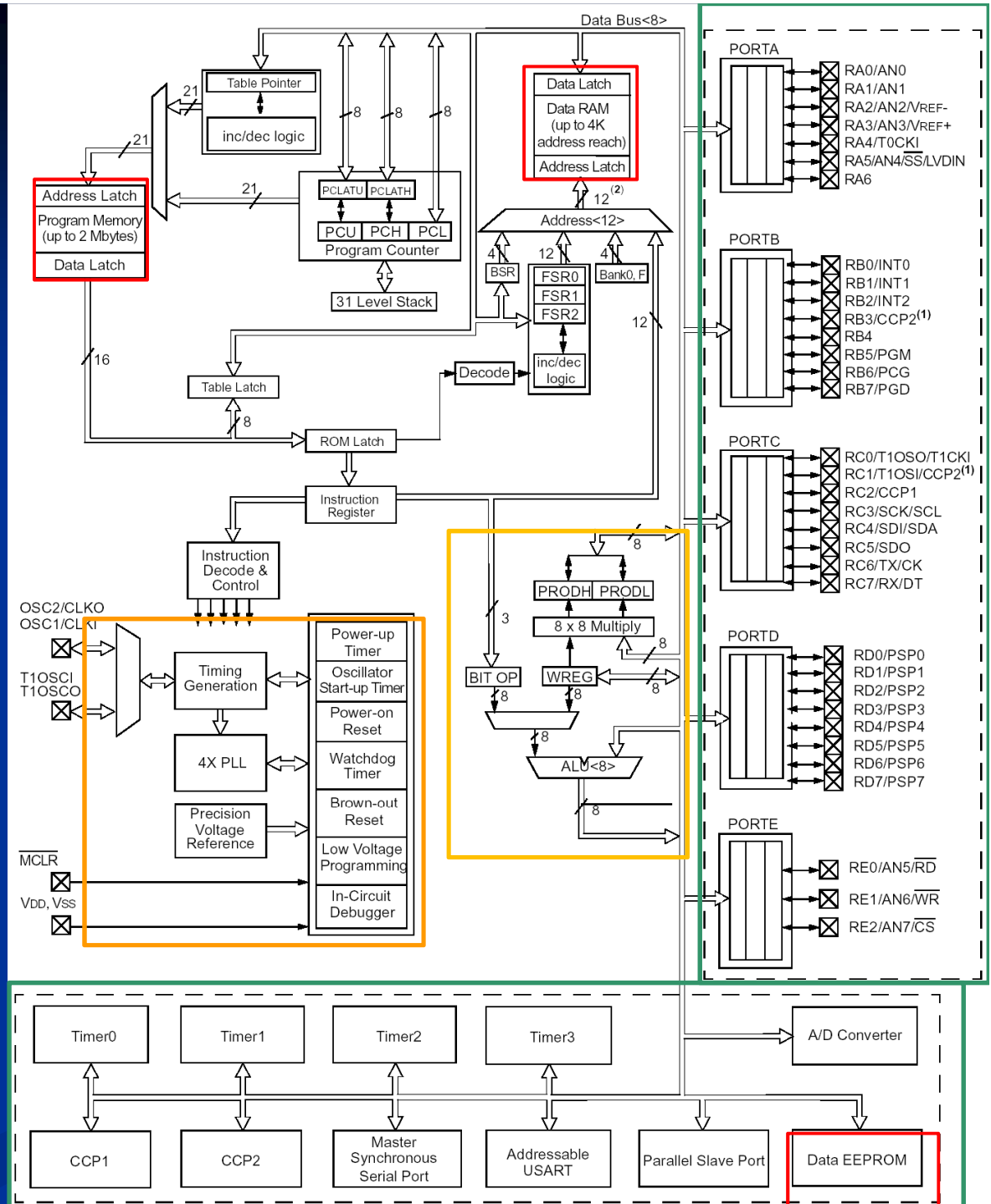
RAM : données volatiles

E2PROM : données
NON volatiles

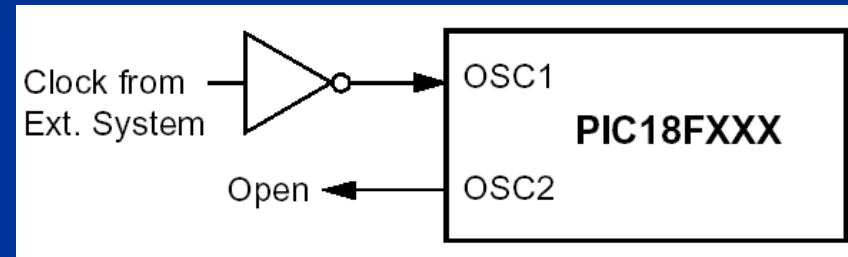
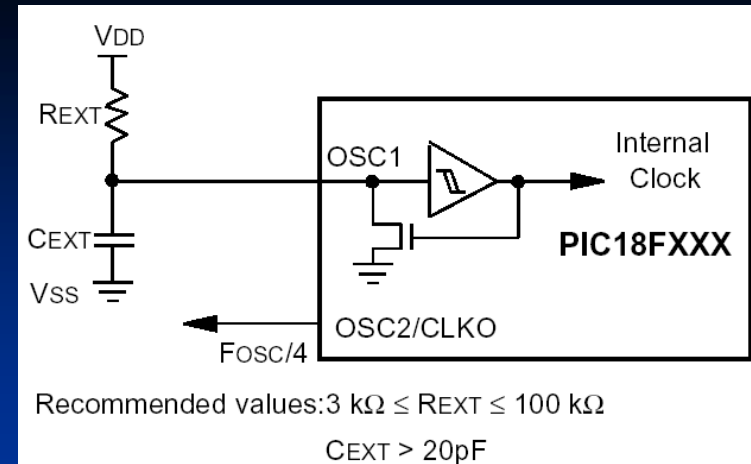
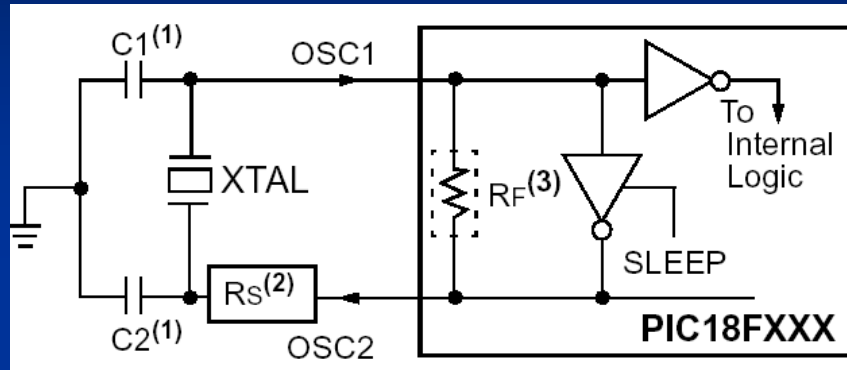
ALU

Horloges et RESET

Périphériques

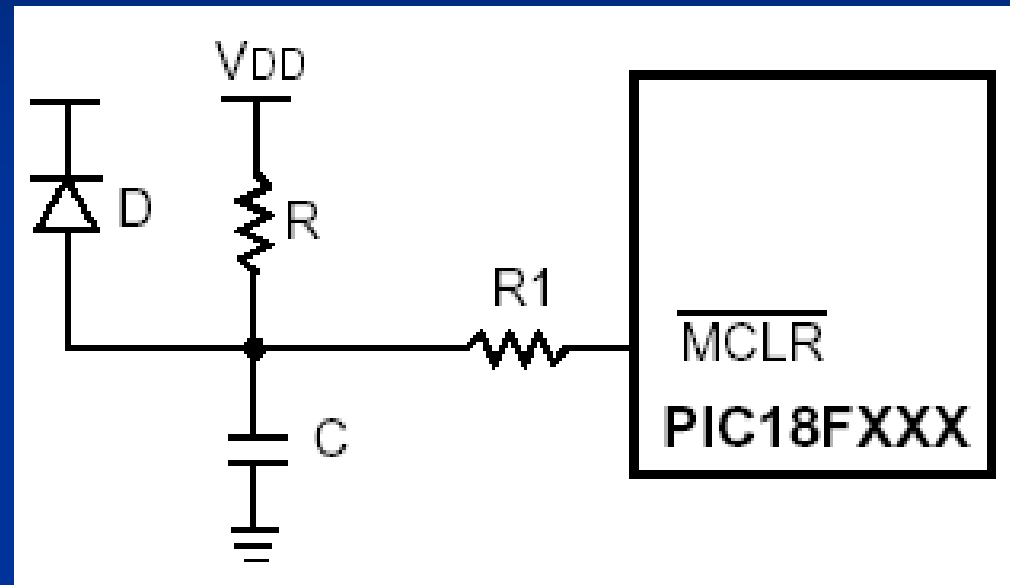


Horloges



Address	Value	Category	Setting
300001	FA	Oscillator	HS
300002	FF	Osc. Switch Enable	RC-OSC2 as RA6
		Power Up Timer	HS-PLL Enabled
		Brown Out Detect	EC-OSC2 as RA6
		Brown Out Voltage	EC-OSC2 as Clock Out
300003	FE	Watchdog Timer	RC
		Watchdog Postscaler	HS
300005	FF	CCP2 Mux	XT
300006	7B	Low Voltage Program	LP
		Background Debug	Enabled

RESET - circuit

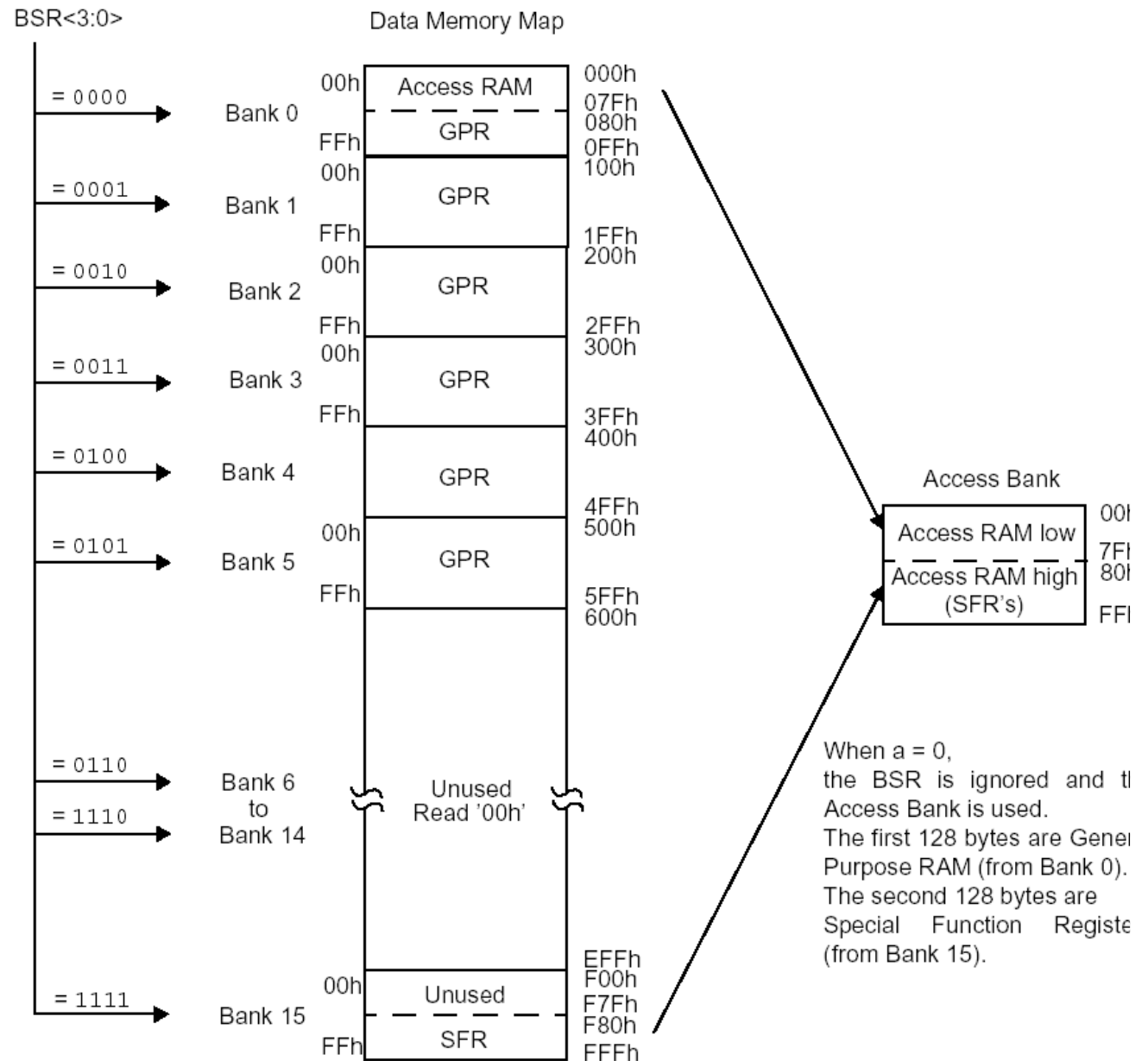


D permet une décharge plus rapide de C lorsque VDD descend.

$R < 40\text{KO}$, $C = 01\mu\text{F}$

$100 < R1 < 1\text{KO}$, R1 limite I dans /MCLR en cas de décharge électrostatique (ESD)

P18Fxxxx - RAM



When a = 1, the BSR is used to specify the RAM location that the instruction uses.

STATUS

Le registre **STATUS** (0xFD8)

7	6	5	4	3	2	1	0
-	-	-	N	OV	Z	DC	C

N si négatif

OV s'il y a eu un débordement dans une opération en complément à 2

Z : si le résultat est nul

DC : demi retenue (le bit4 est passé à 1)

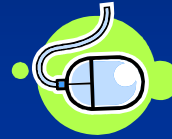
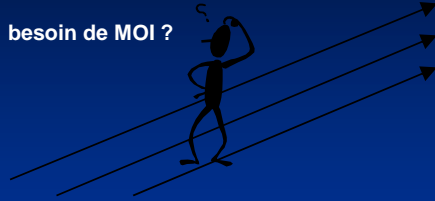
C : s'il y a eu une retenue (résultat supérieur à 0xFF)

INTERRUPTIONS

Scrutation VS Interruptions

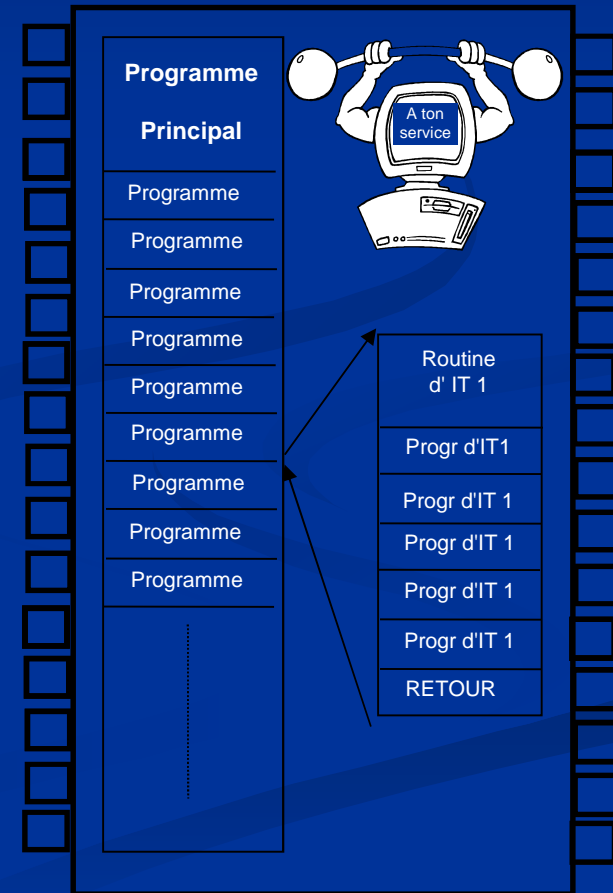
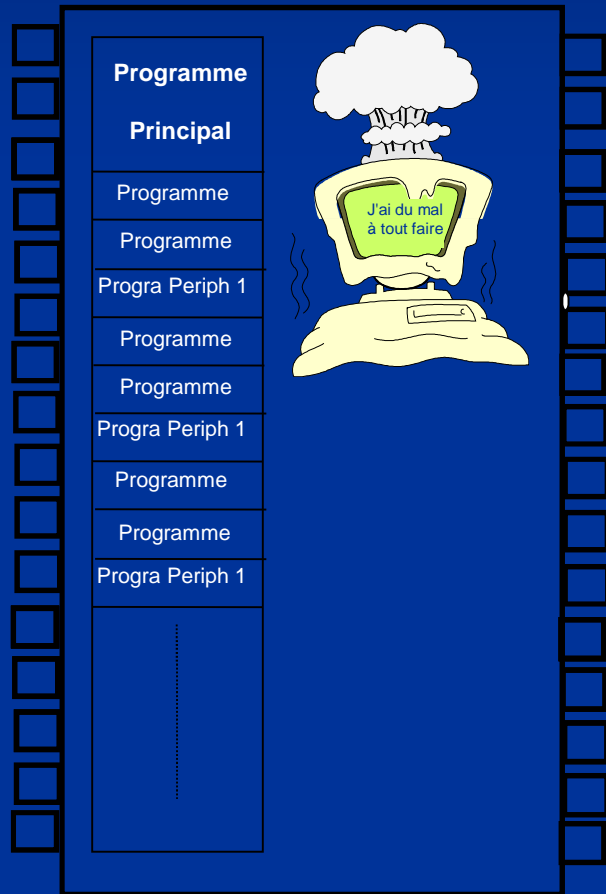
○ Scrutation - Polling

A-t-il besoin de MOI ?

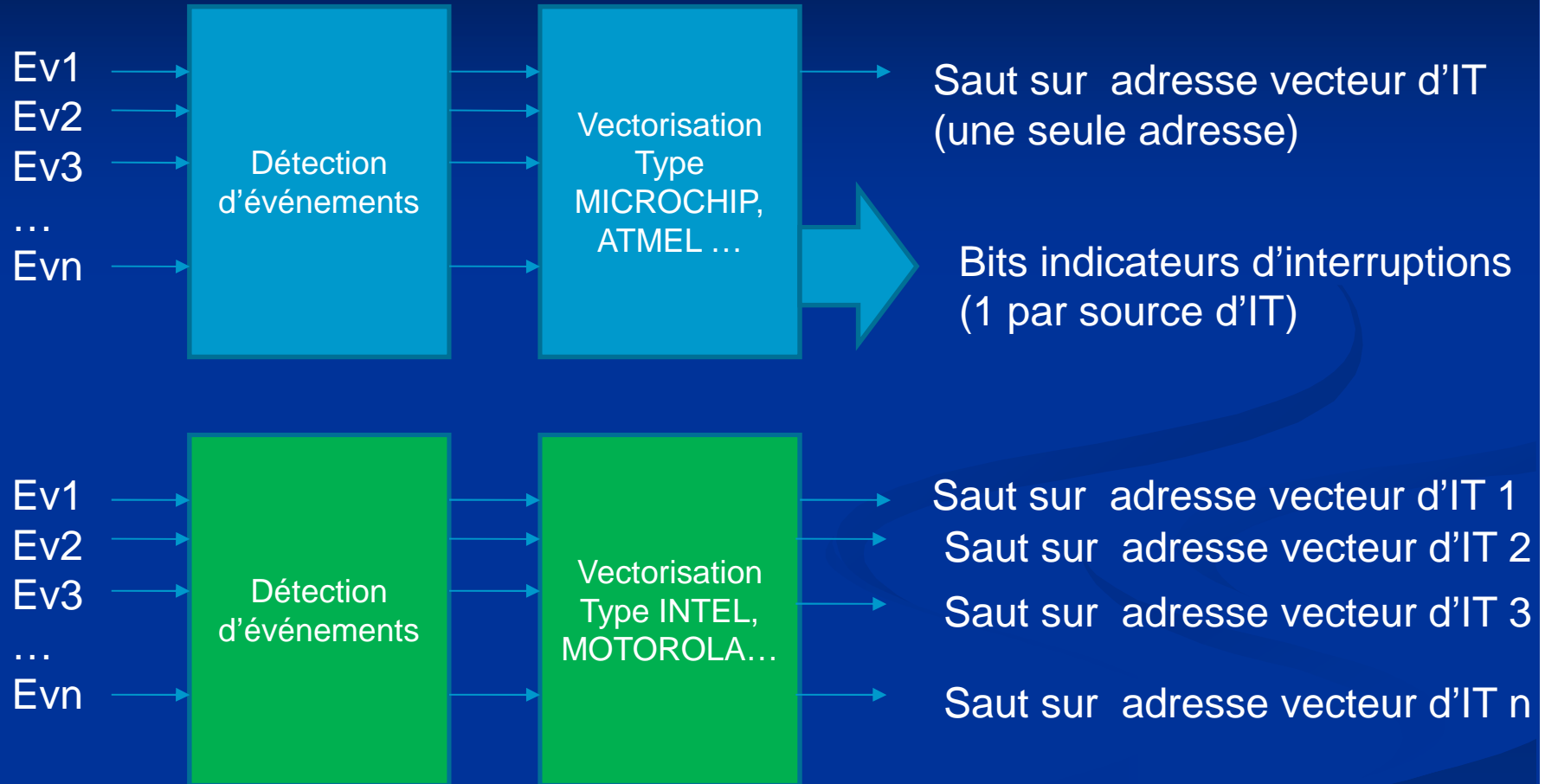


○ Interruption

J'ai besoin de TOI !!!



Deux approches



interruptions

- **Validation globale**

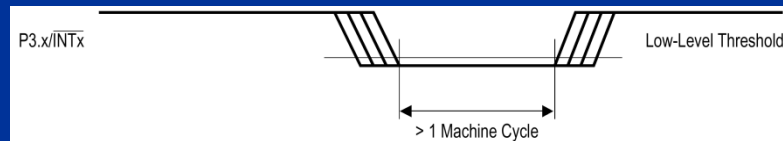
Inhibition ou Validation de toutes les interruptions

- **Niveau de priorité**

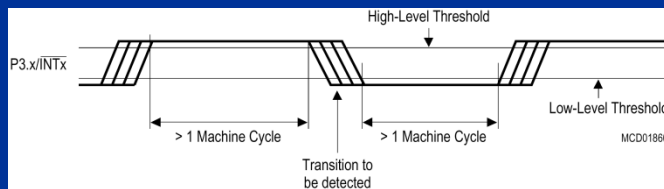
Interruptible si priorité N+1

- **Type de déclenchement**

- **Sur niveau**



- **Sur front**



- **Routine d'interruption**

Sous-programme d'IT

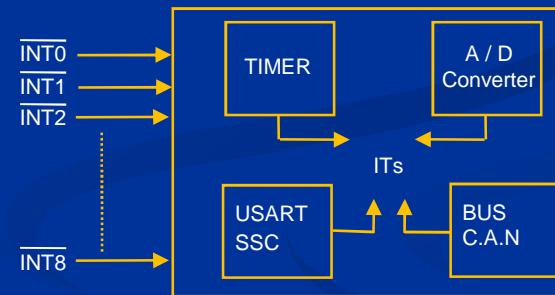
- **Masquable**

Une interruption masquable peut être mise hors service par le microcontrôleur

- **Non Masquable**

Une interruption non masquable (NMI) ne peut être ignorée

- **Externe, Interne**



- **Acquittement**

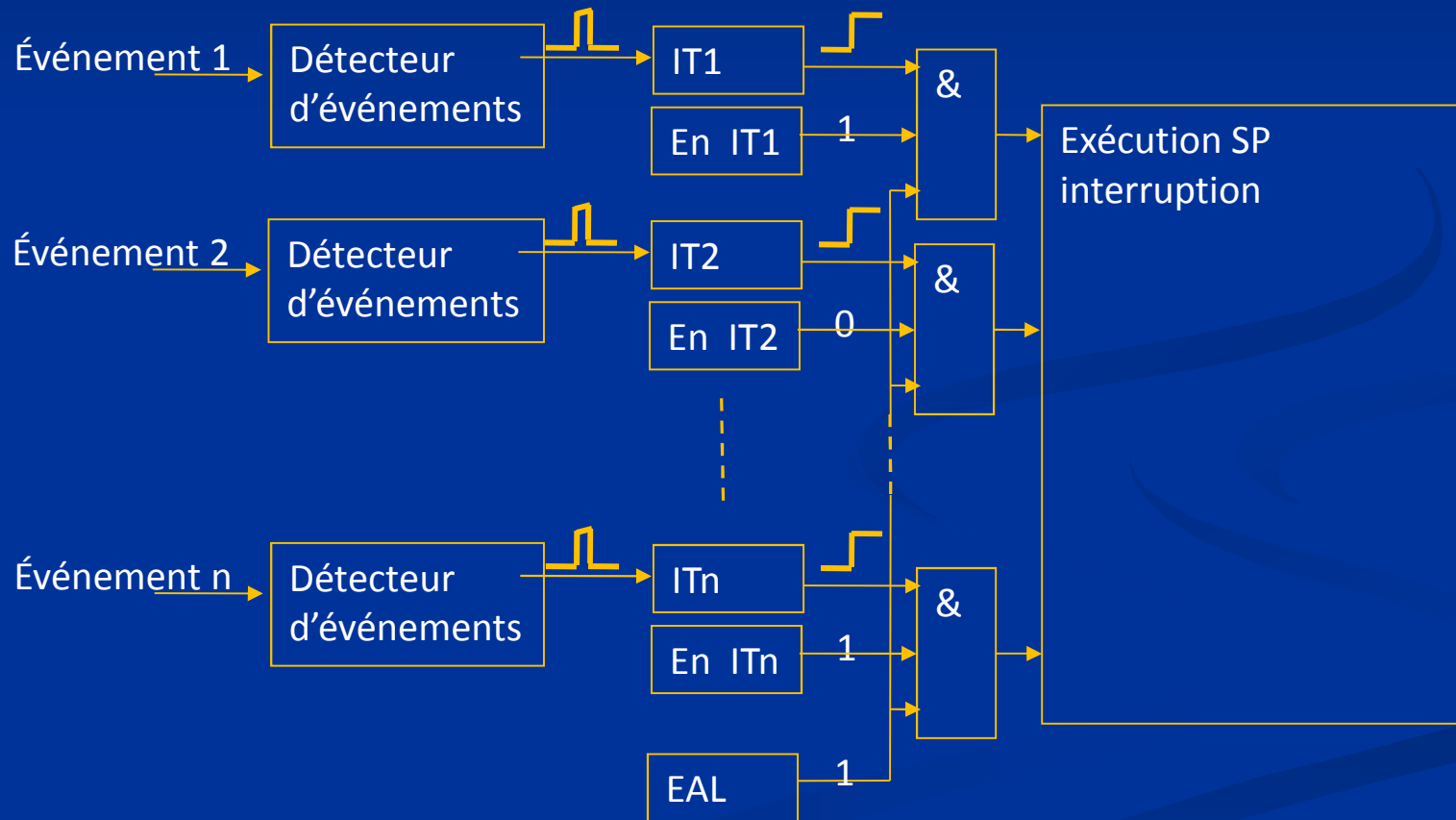
- **Matériel**

- **Logiciel**

- **Vecteur**

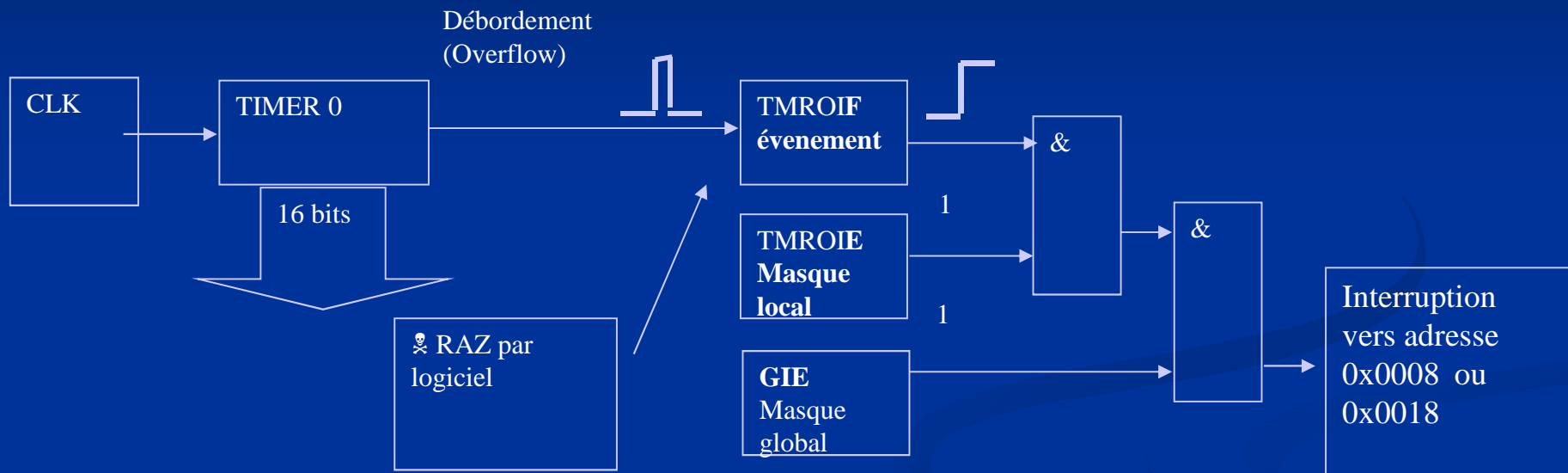
N° Vecteur ----> Vecteur

Masquage / Validation

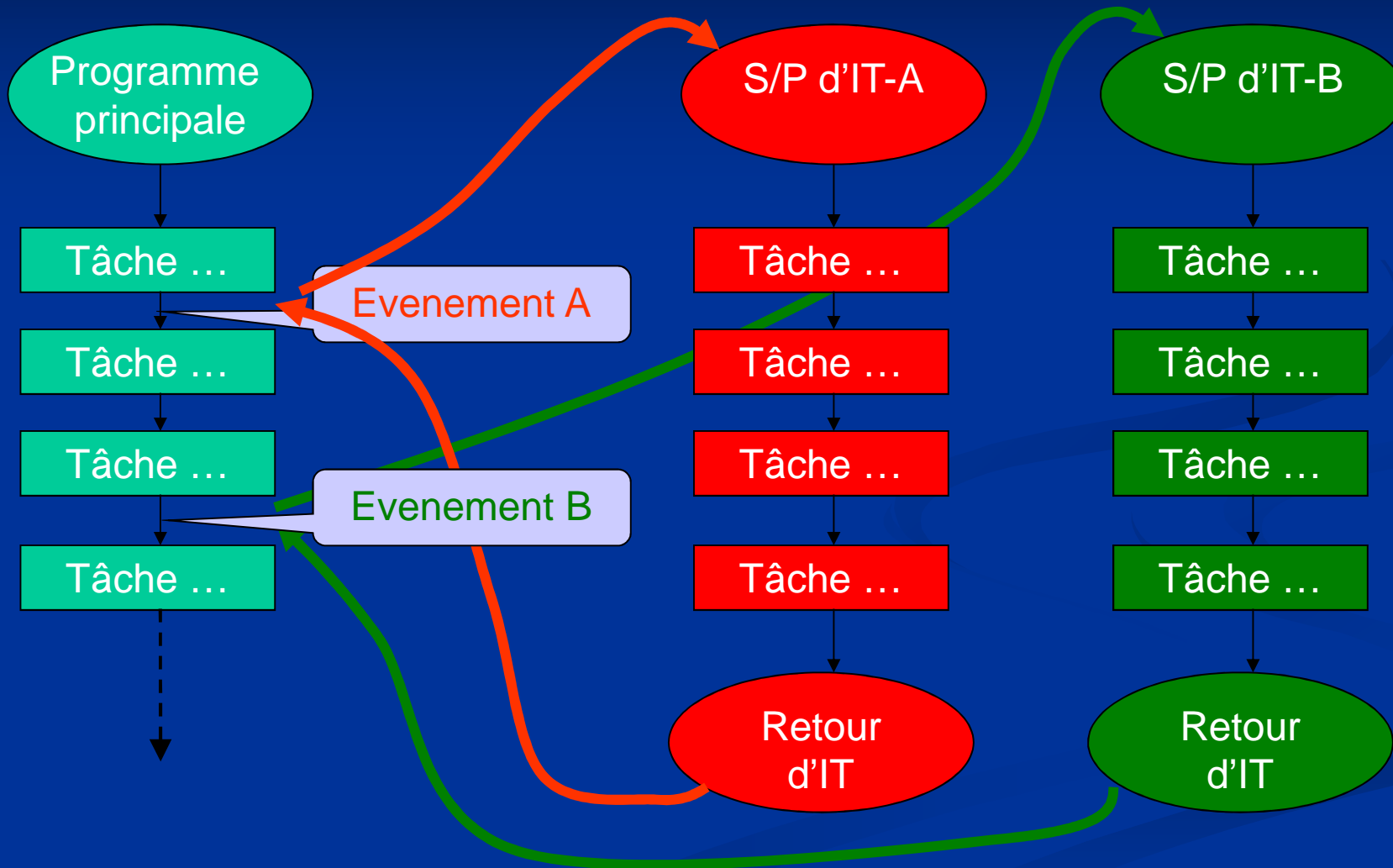


Interruptions- principes

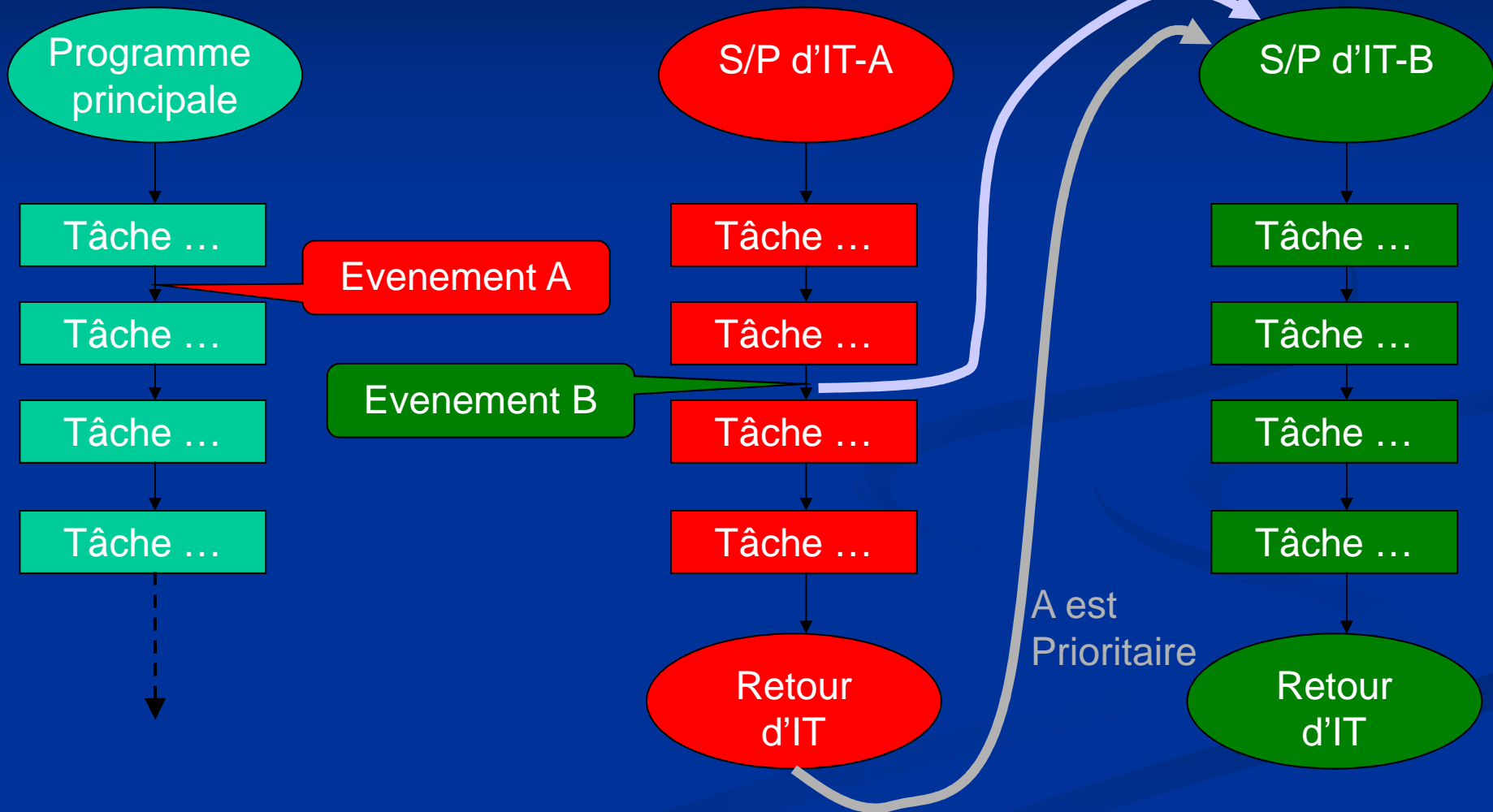
TIMER0



Interruptions – Priorités (1)



Interruptions – Priorités (2)



Choix des priorités RCON et INTCON

REGISTRE RCON : IPEN : interrupt priority enable. Cette fonction peut être désactivée pour avoir une compatibilité

logicielle avec l'unité centrale PIC16.

7	6	5	4	3	2	1	0
IPEN	-	-	RI	TO	PD	POR	BOR

Registre INTCON:

GEIH, GEIL de INTCON si IPEN=1

GEIH : global interrupt enable high (validation des interruptions prioritaires, adresse 0x0008)

GEIL : global interrupt enable low (validation des interruptions non prioritaires, adresse 0x0018)

7	6	5	4	3	2	1	0
GEIH	GEIL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

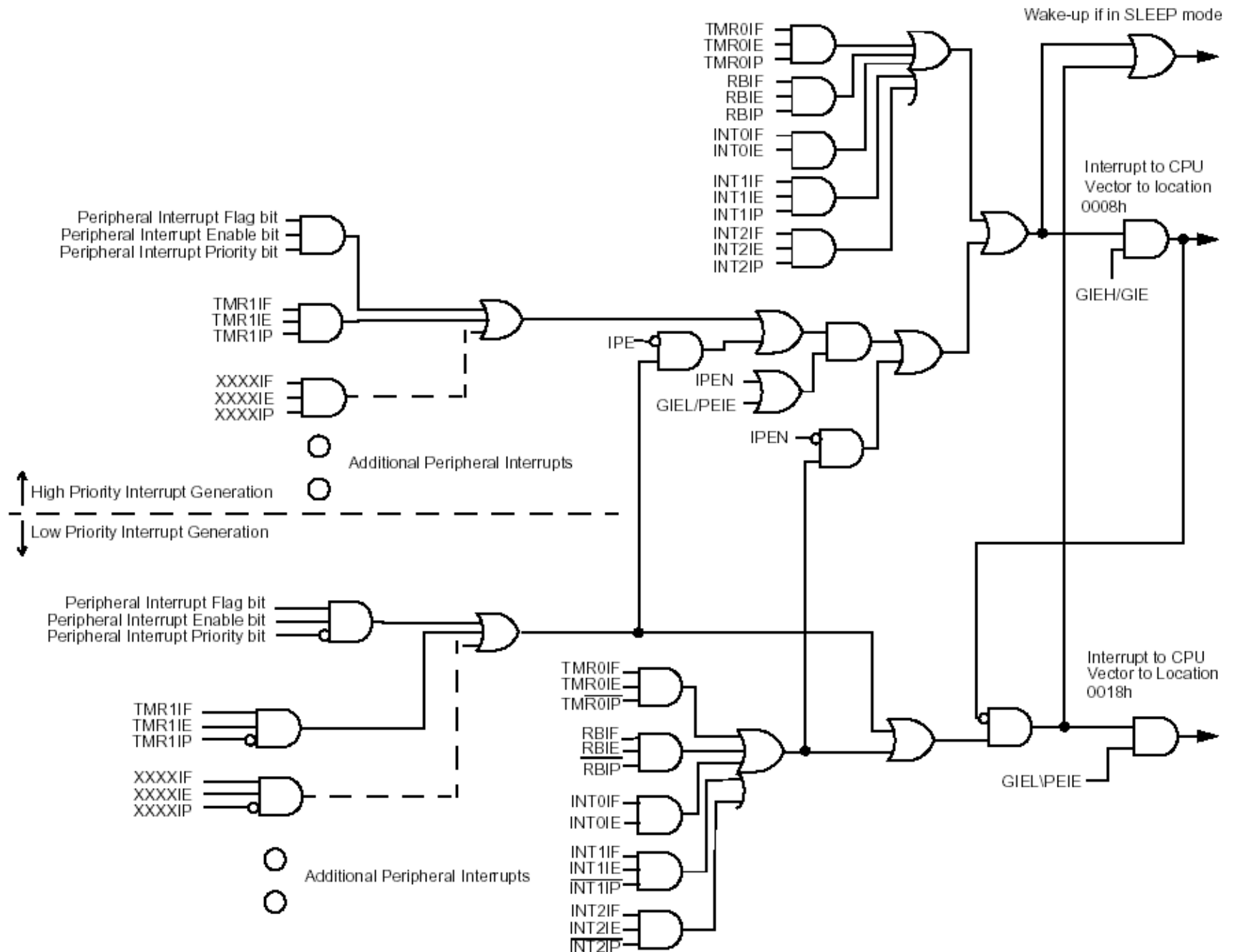
Adresses des SP d'IT

Il n'y a que 16 instructions insérables entre les adresses 0x0008 et 0x0018.

Il faudra pour les interruptions prioritaires (0x0008) placer à cette adresse un saut vers la SP d'IT. (goto)

Pour les interruptions non prioritaires le SP d'IT pourra être placé à l'adresse 0x0018

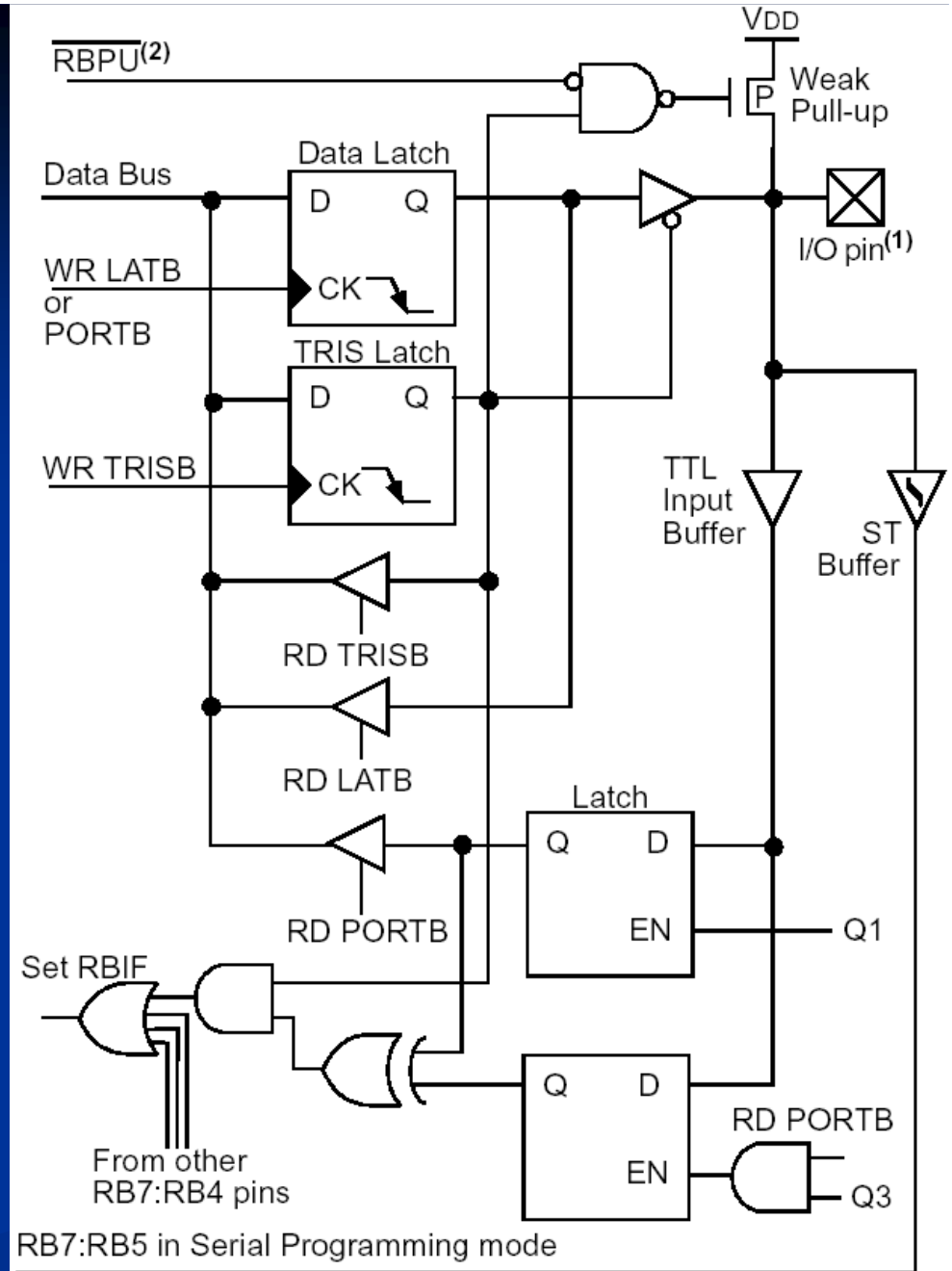
Interruptions - généralités



Périphériques des PIC18

Ports parallèles (PRB)

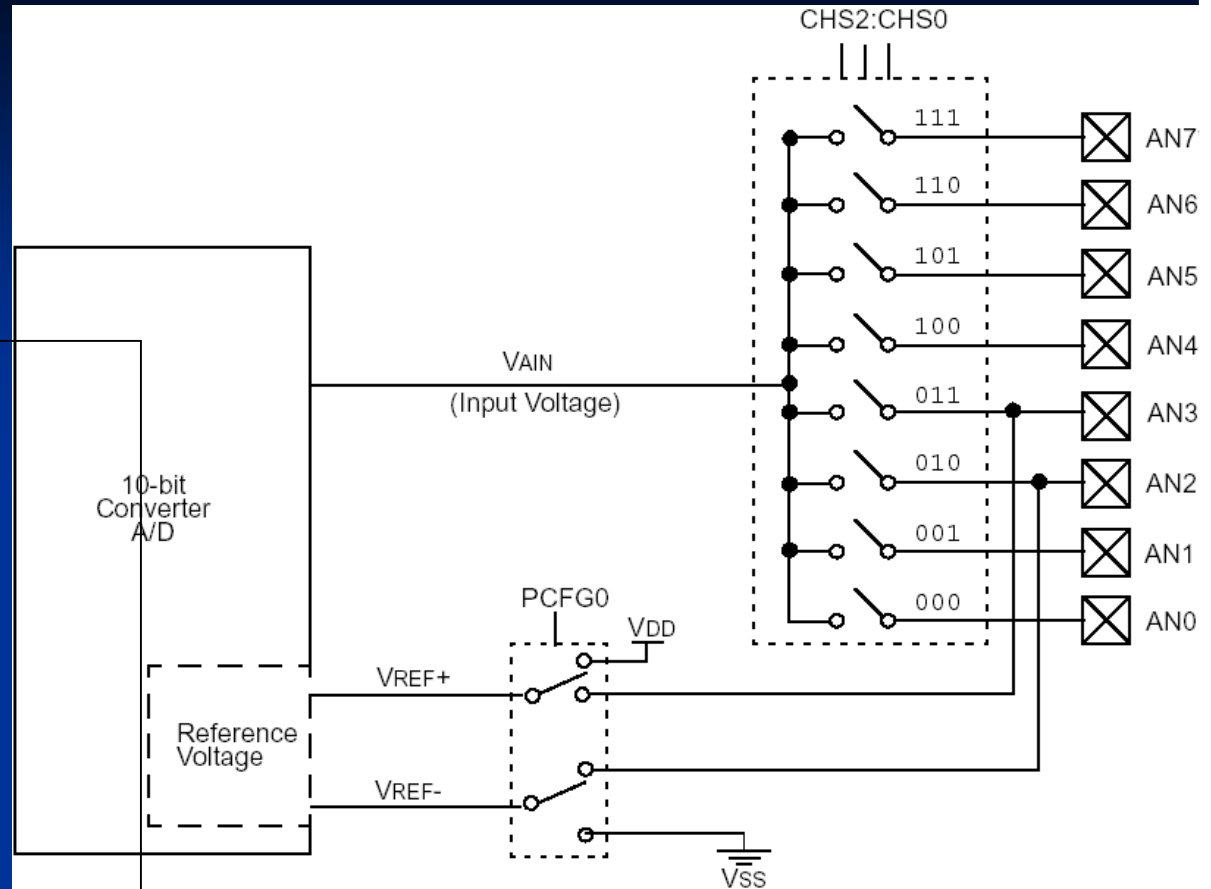
```
#include  
<p18f452.h>  
void main(void)  
{  
  char a=0,b=0x55 ;  
  PORTB=0 ;  
  TRISB=0b11110000 ;  
  
  a=PORTB ;  
  PORTB=b ;  
  While(1) ;  
}
```



ADC

```
// Demo pour ADC
#include <p18f452.h>
#define q 4.8828e-3// quantum

void main(void)
{
float res;
// CAN on. CLOCK=FOSC/2. CANAL0
// seul AN0 est activé
// VREF+=VDD VREF-=VSS
ADCON0=1;
ADCON1=0x8E;
while(1){
// déclenche SOC
ADCON0bits.GO_DONE=1;
// attend EOC
while(ADCON0bits.GO_DONE);
// calcule la tension
res=(float)ADRES*q;
}
}
```



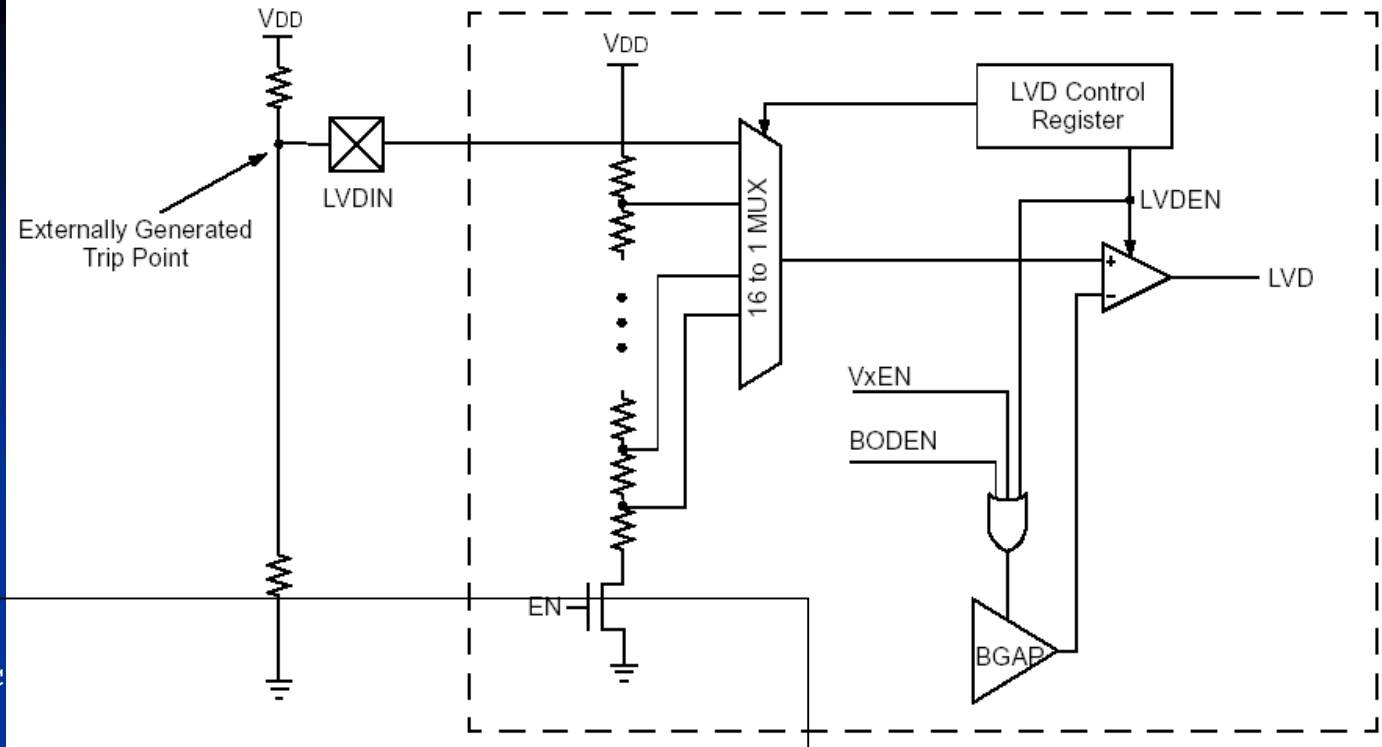
REGISTRE ADCON0

7	6	5	4	3	2	1	0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON

REGISTRE ADCON1

7	6	5	4	3	2	1	0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0

LVD



```

// Demo pour LVD
#include <p18f452.h> c
void main(void)
{
// LVD active, pas d'IT
// detection VDD<3.5v
PIE2bits.LVDIE=0;
LVDCON=0b00011001;
While(!LVDCONbits.IRVST); //attend...
    while(1){
        if (PIR2bits.LVDIF)
        {
            PIR2bits.LVDIF=0
        }
        else test=0;
    }
}

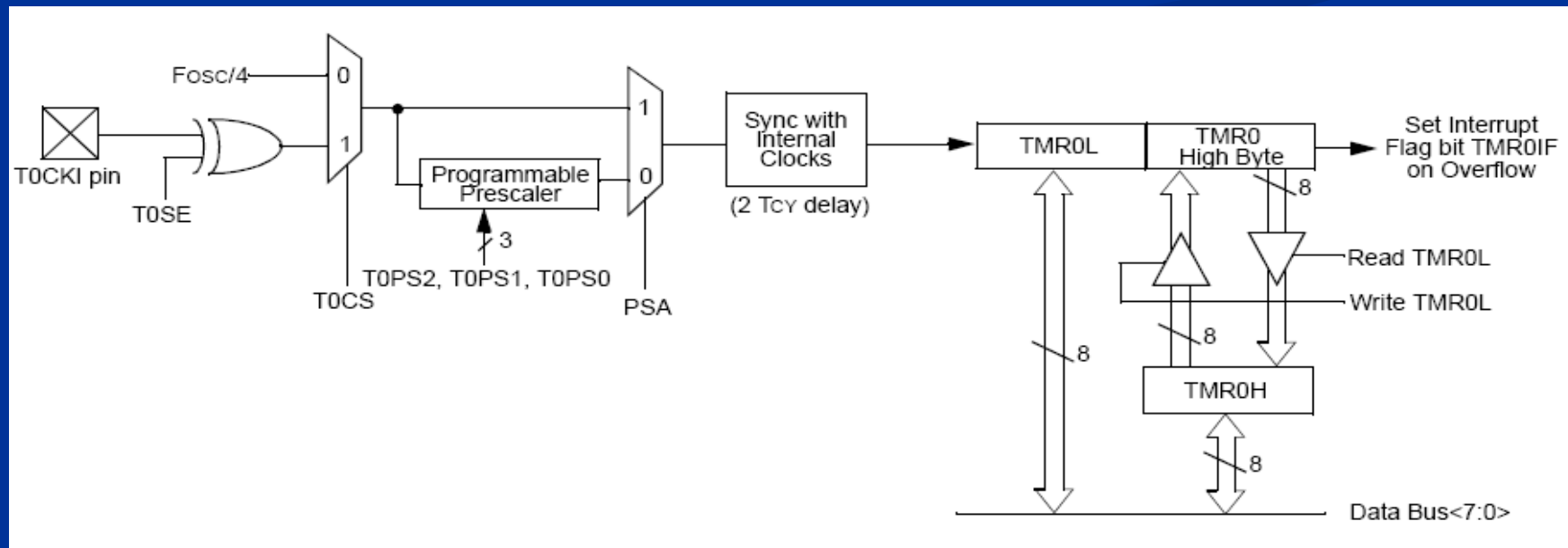
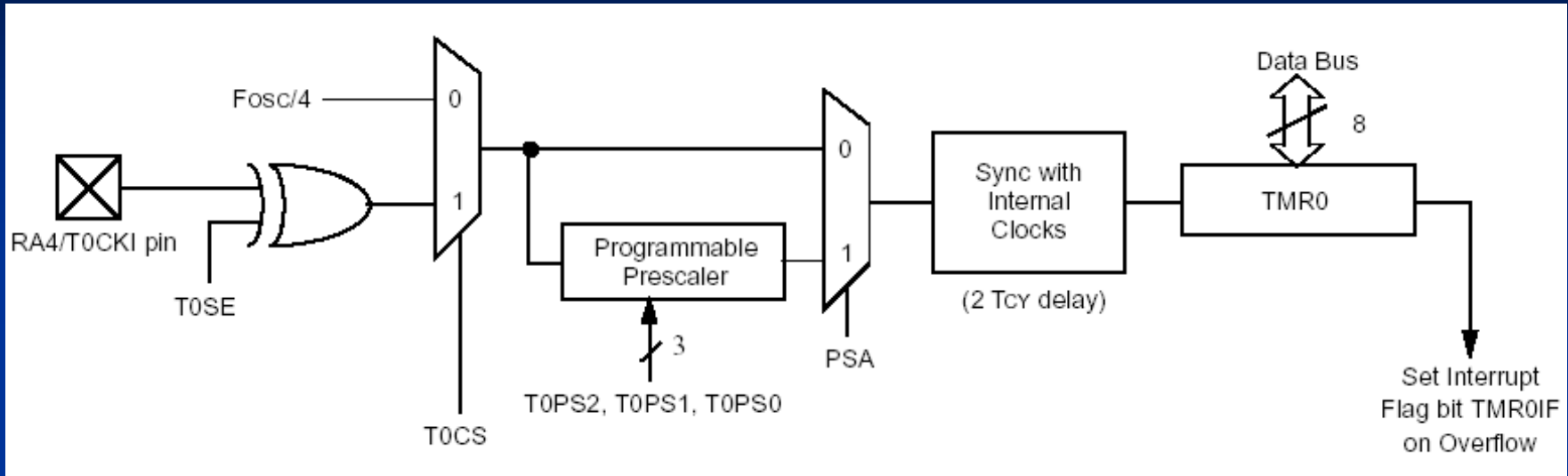
```

7	6	5	4	3	2	1	0
—	—	IRVST	LV DEN	LV DL3	LV DL2	LV DL1	LV DL0

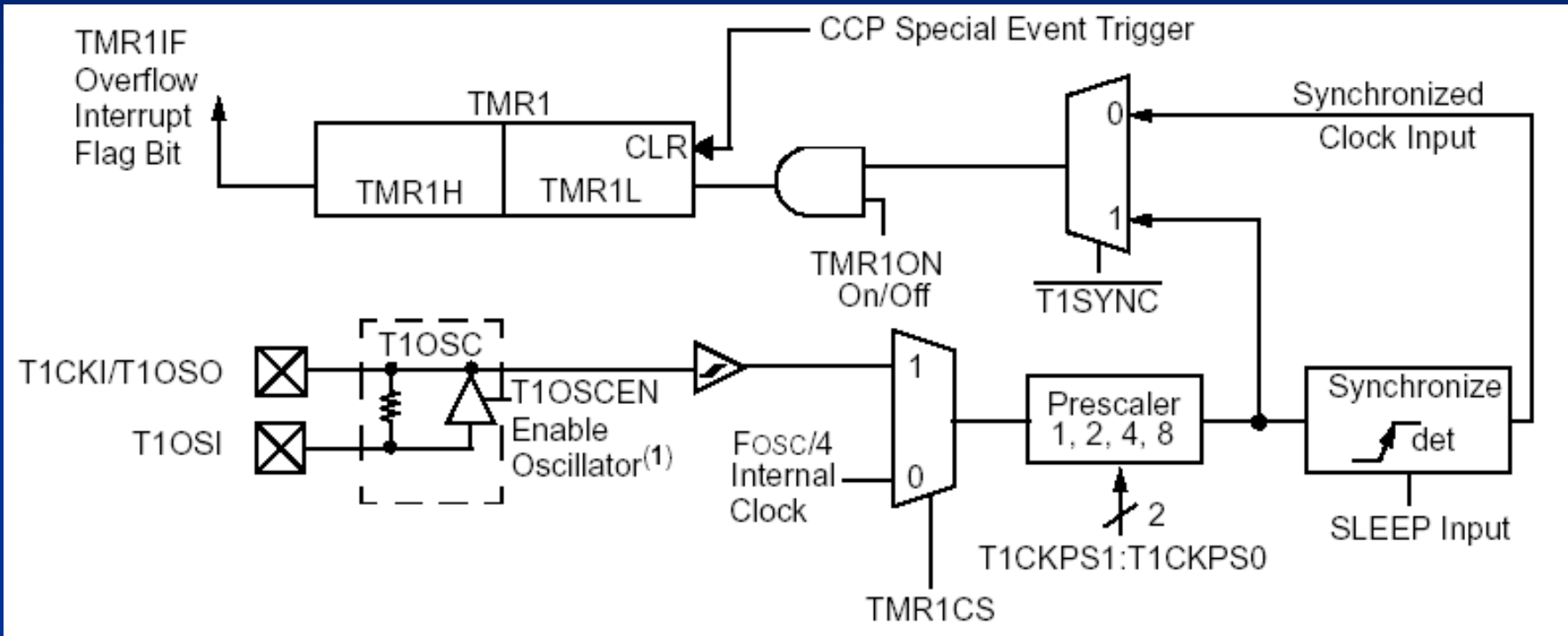
REGISTRE **LVDCON** (0xFD2)

test=1;

TIMER0



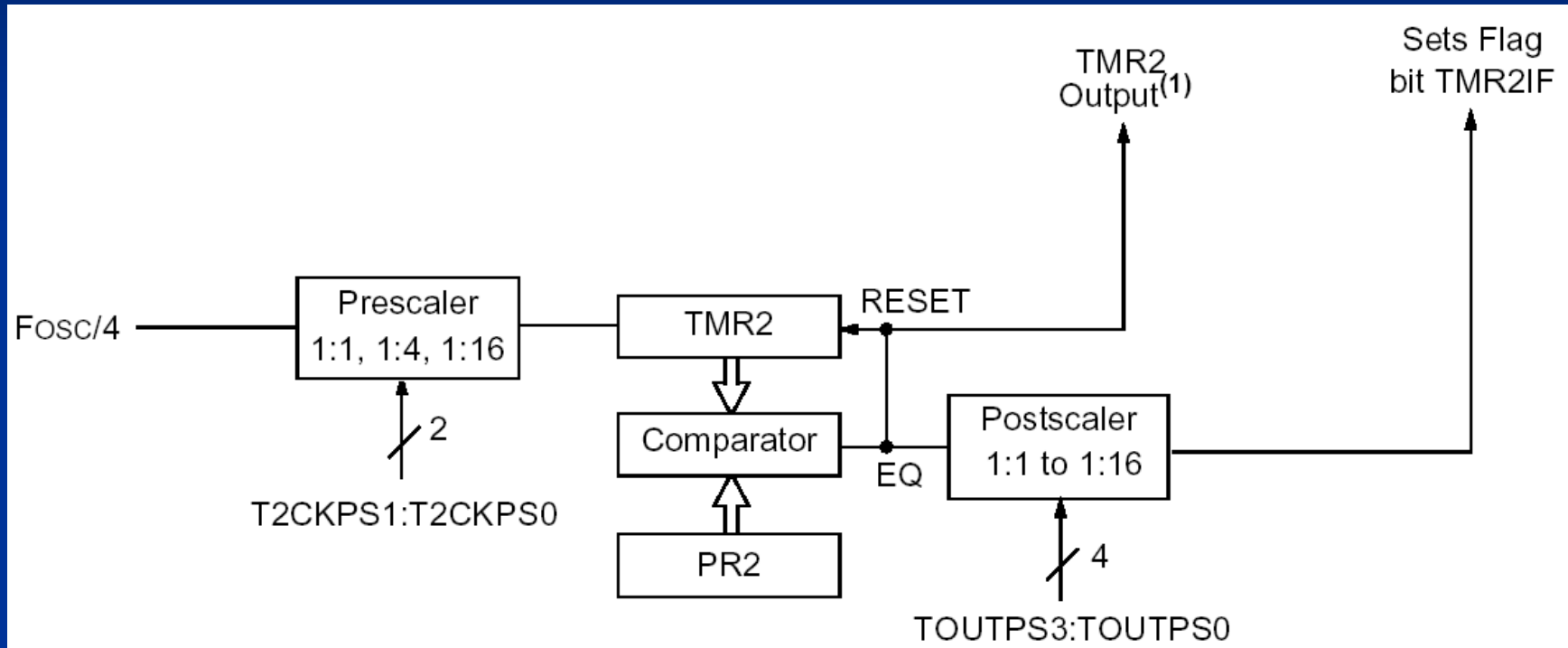
TIMER1



REGISTRE T1CON

7	6	5	4	3	2	1	0
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

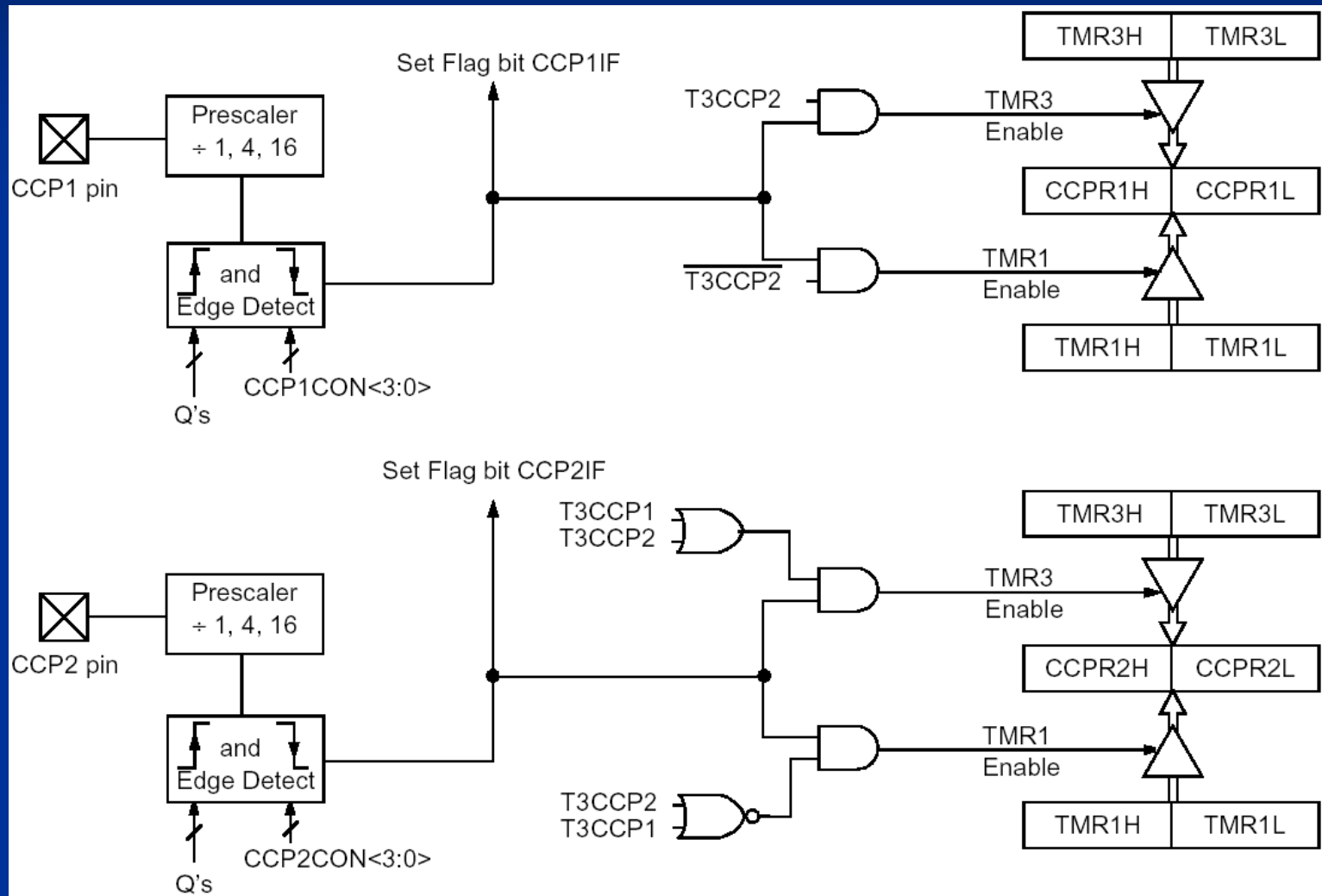
TIMER2



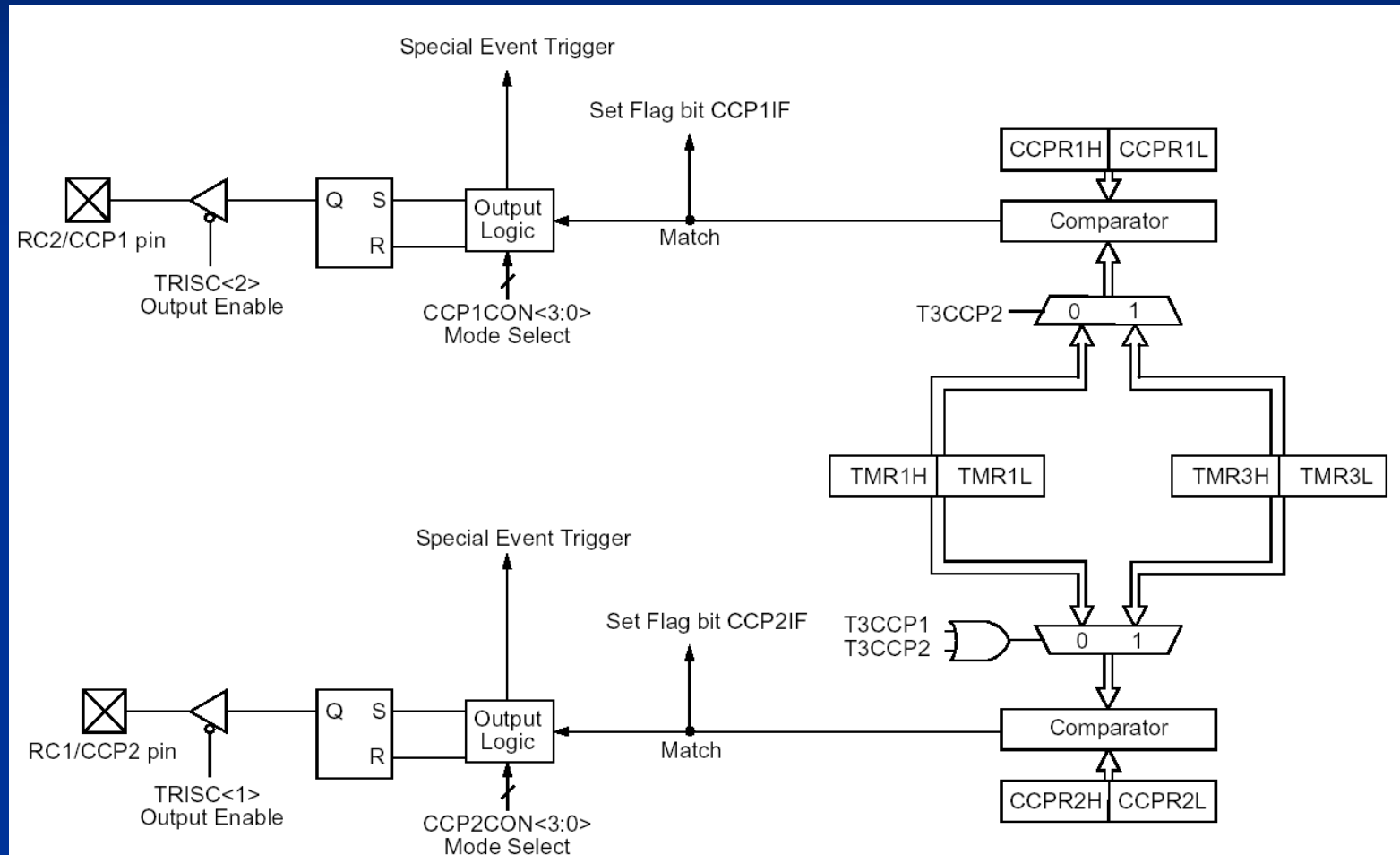
REGISTRE TCON2

7	6	5	4	3	2	1	0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

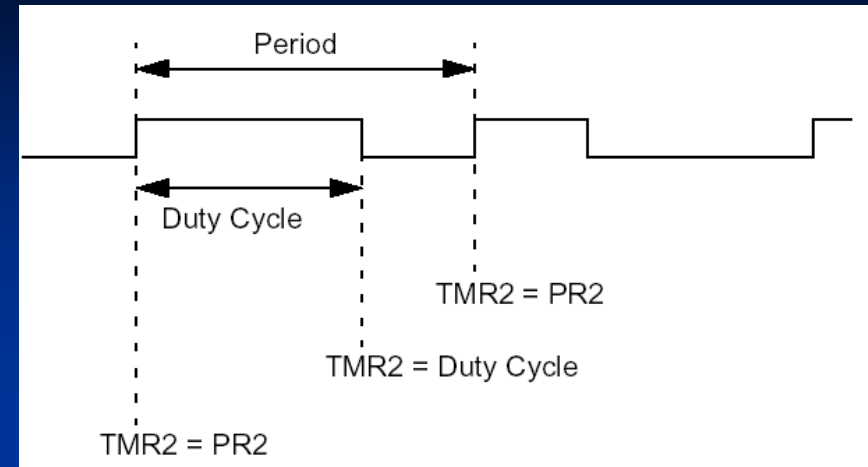
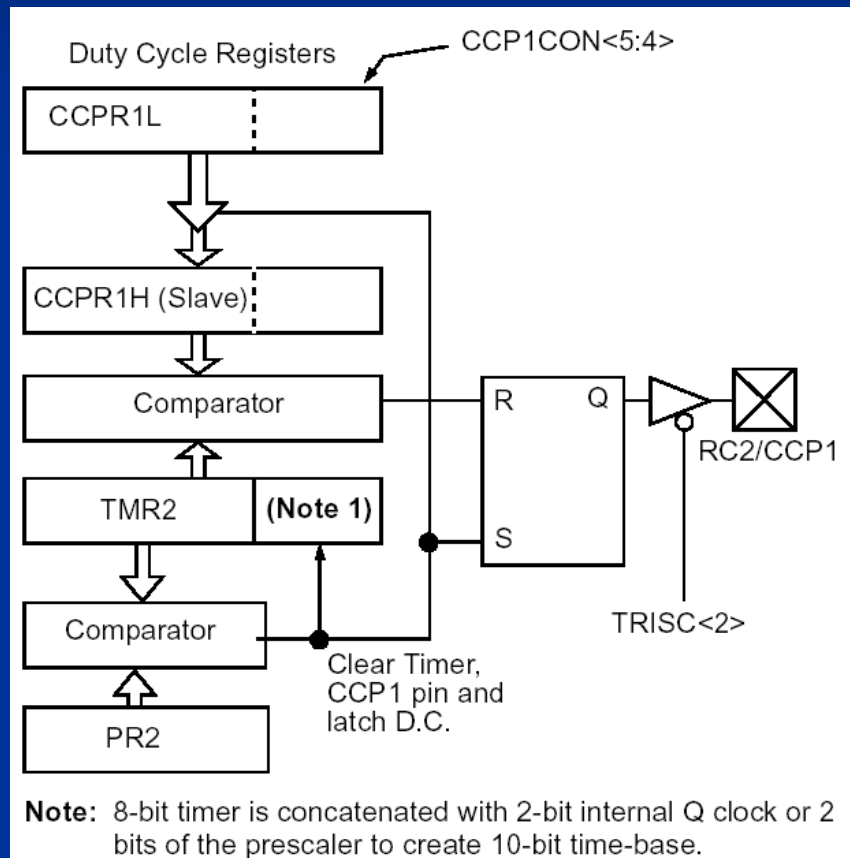
CAPTURE



COMPARE



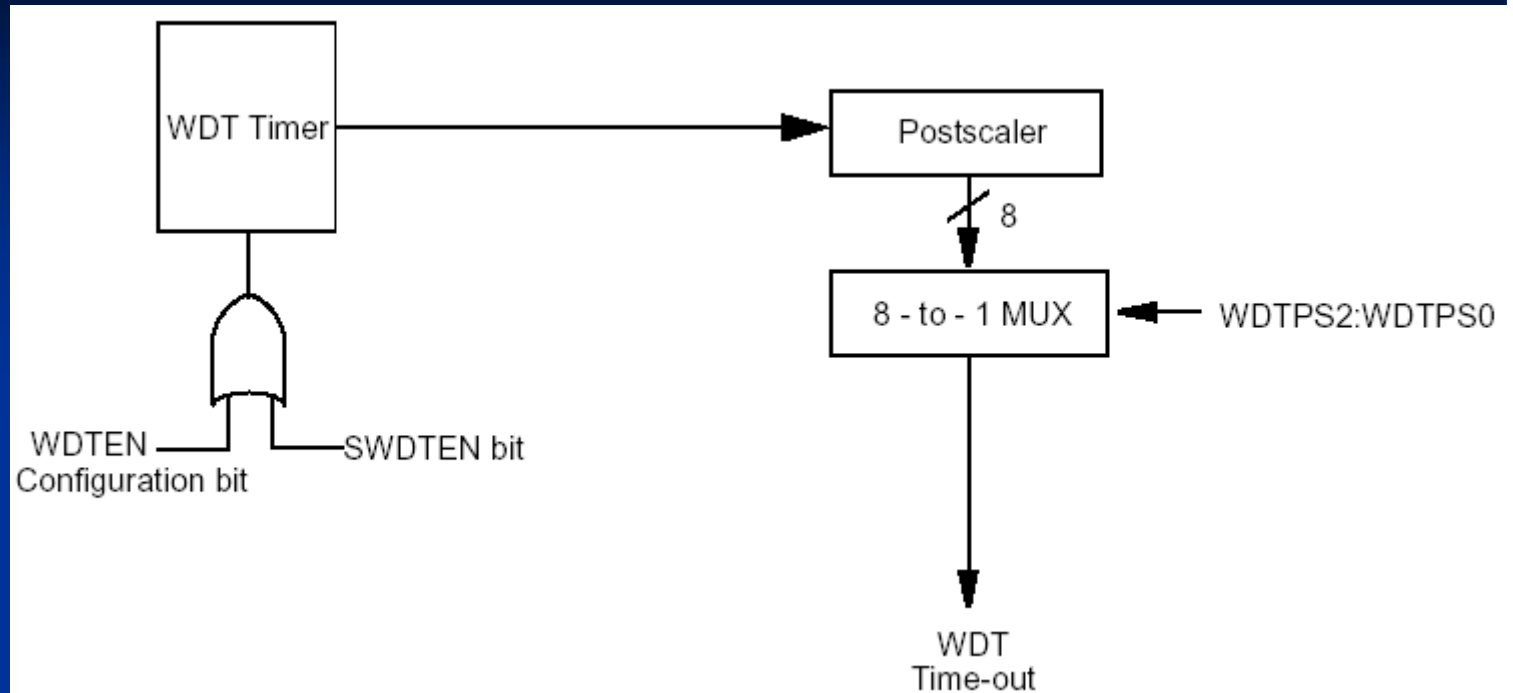
PWM



CCP1=0
 Lorsque TMR2=PR2
 CCP1=1
 TMR2=0
 CCPR1H=CCPR1L
 Lorsque TMR2=CCPR1H
 CCP1=0
 Lorsque TMR2=PR2
 Etc...

PR2 représente la période T
 CCPR1L représente th

Watch Dog (WD)



REGISTRE CONFIG2H

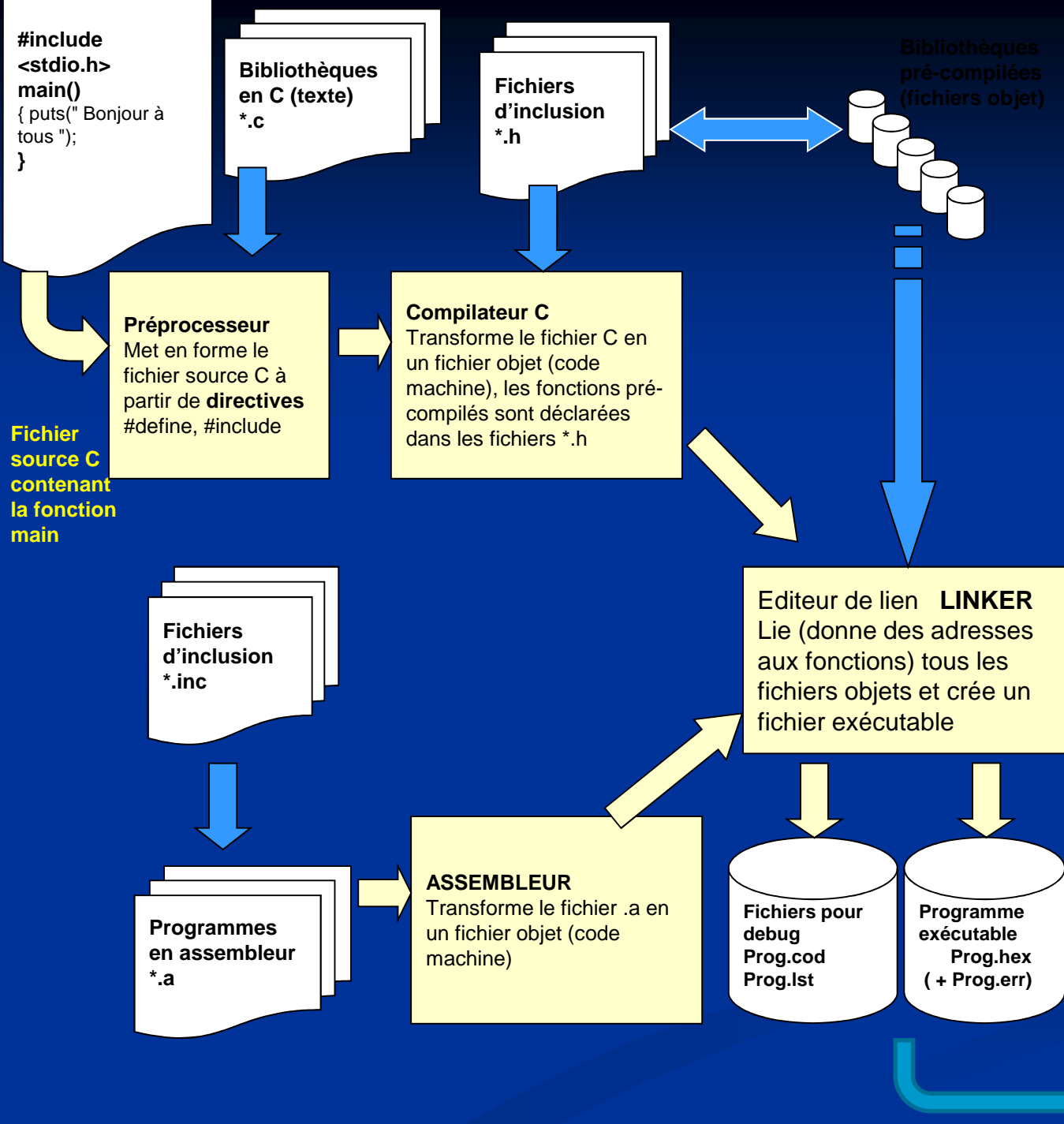
7	6	5	4	3	2	1	0
—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN

Les bits WDTPS2-WDTPS0 représente le rapport de division de la sortie WDT TIMER (1 à 8)
 Après activation le chien de garde doit être réinitialisé avant la génération du Time-Out qui provoque un RESET
 Les instructions assembleur « clrdwt » et « sleep » remettent le TIMER à 0.
 Durée de comptage avant Time-Out et sans prédiviseur $7\text{mS} < T < 33\text{ mS}$
 Avec un prédiviseur de 5 : $35\text{mS} < T < 165\text{mS}$

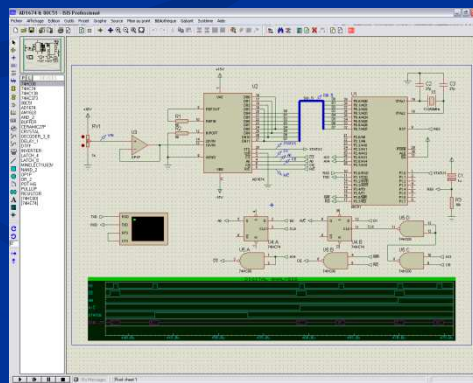
OUTILS DE DEVELOPPEMENT



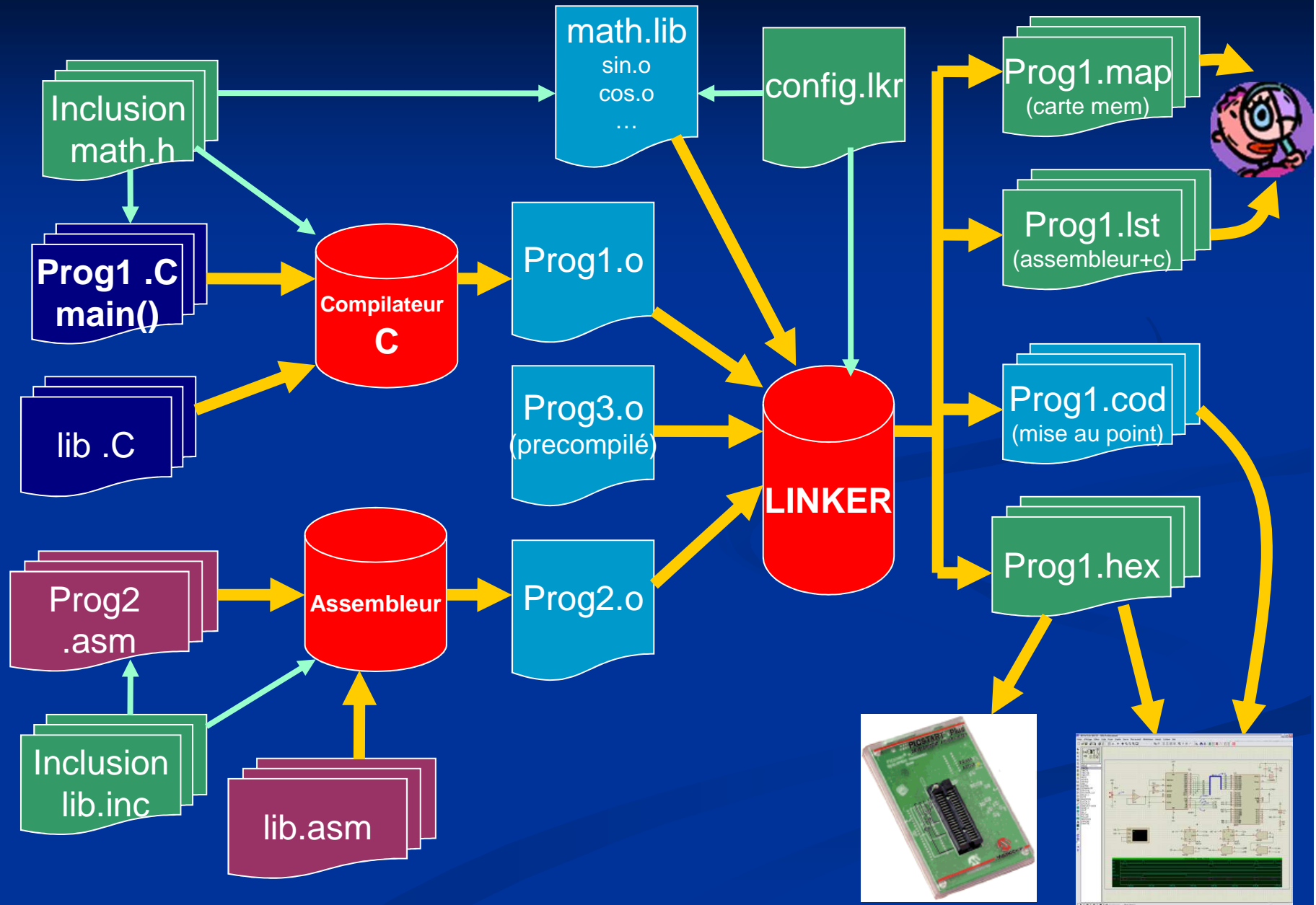
Outils de développement



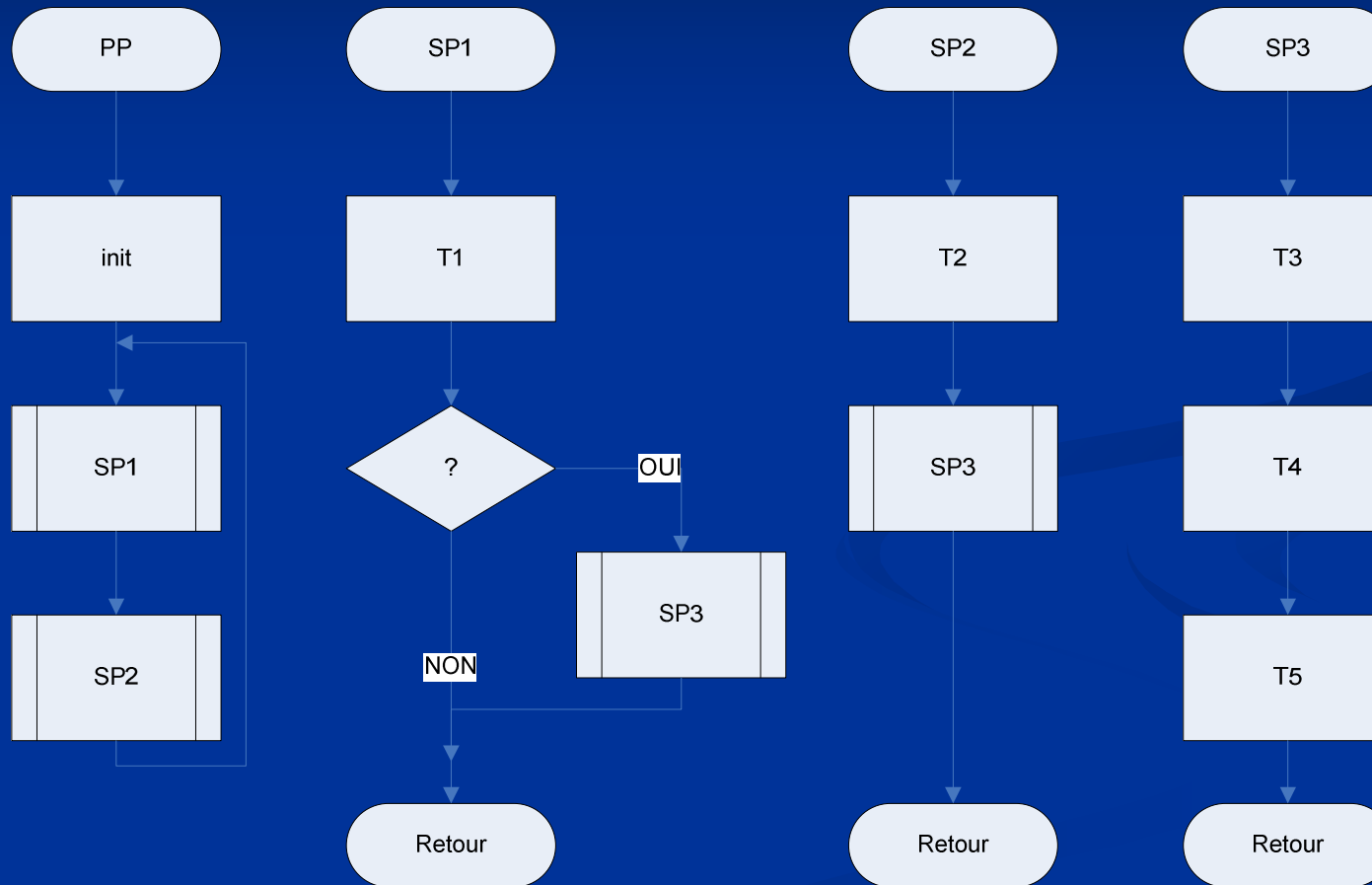
Fichier source C contenant la fonction main



Flux des données



Programmation structurée



FIN (provisoire)

