S34PH412 : Systèmes Microprogrammés

Architecture des Ordinateurs

Cours 7

Université de la Réunion

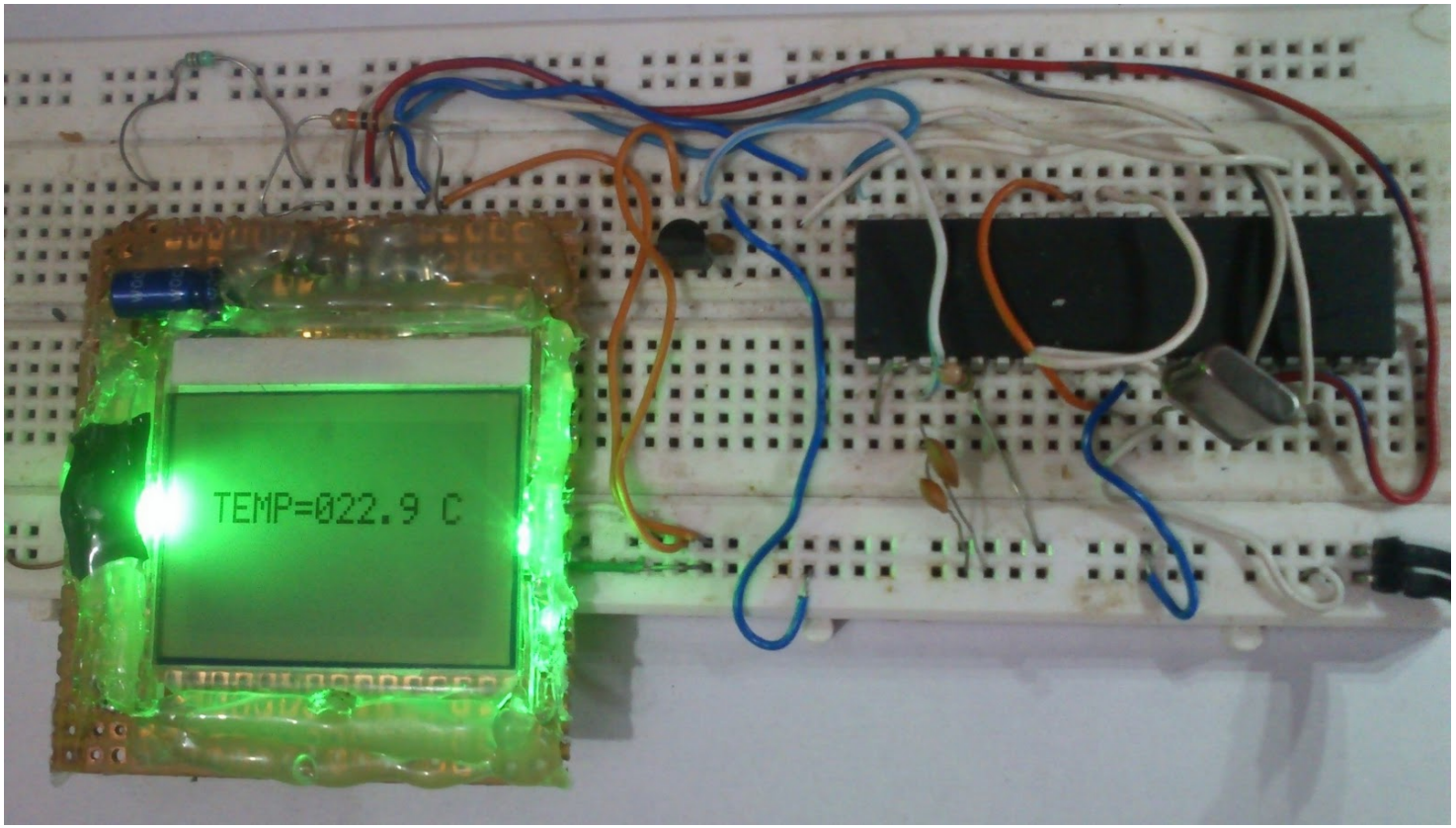
Année 2015/2016

Alicalapa F.

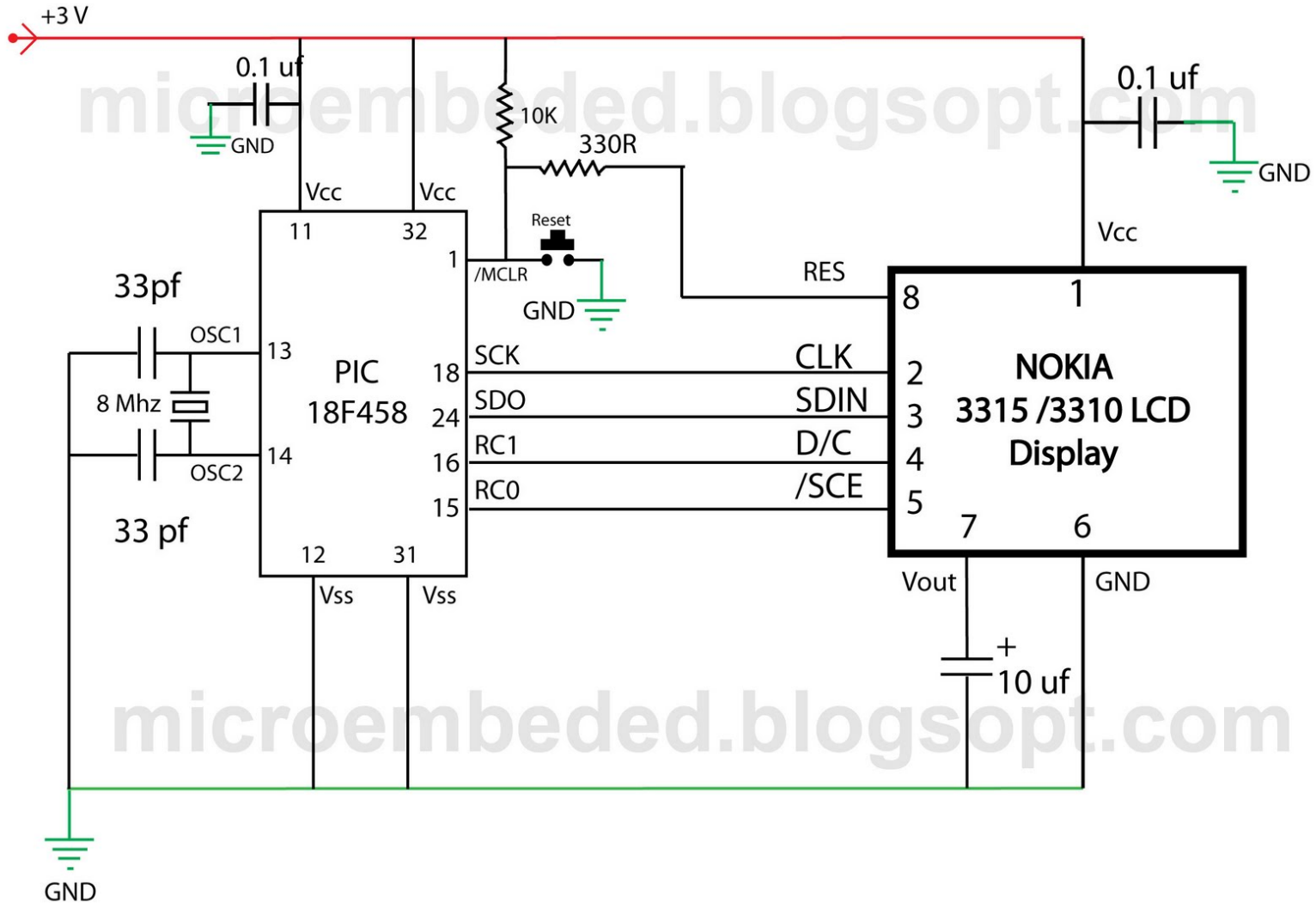
UNIVERSITÉ DE LA RÉUNION
S'ouvrir aux mondes

UFR Sciences et Technologies UNIVERSITÉ DE LA RÉUNION

- ETUDE DE CAS 1 CM7 : écran Nokia 3315 LCD, PIC uC
  - **Notions à découvrir : découplage alimentation, RESET double ICI, Quartz**



- ETUDE DE CAS 3 :
  - **Notions à découvrir : découplage alimentation, RESET double ICI (bien observé le figure), Quartz**



- **systèmes embarqués**

Embedded Systems/ARM Microprocessors

The ARM architecture is a widely used 32-bit RISC processor architecture. In fact, the ARM family accounts for about 75% of all 32-bit CPUs, and about 90% of all embedded 32-bit CPUs. ARM Limited licenses several popular microprocessor cores to many vendors (ARM does not sell physical microprocessors). Originally ARM stood for *Advanced RISC Machines*.

Some cores offered by ARM:

ARM7TDMI

ARM9

ARM11

Some examples of ARM based processors:

Intel X-Scale (PXA-255 and PXA-270), used in Palm PDAs

Philips LPC2000 family (ARM7TDMI-S core), LPC3000 family (ARM9 core)

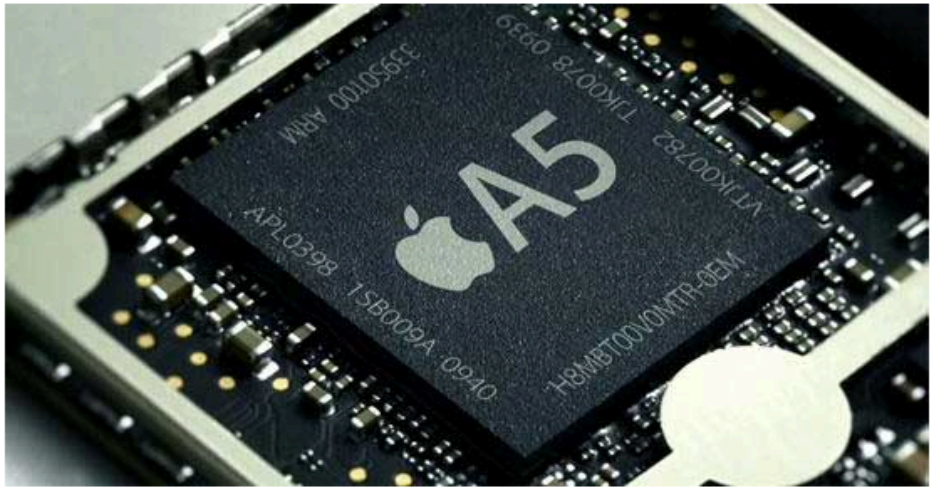
Atmel AT91SAM7 (ARM7TDMI core)

ST Microelectronics STR710 (ARM7TDMI core)

Freescale MCIMX27 series (ARM9 core)

IDC: ARM-Powered Devices, Such As Apple's iPad Can Be Classified a PC Device

Ed Sutherland (6:45 am PDT, May 5th 2011)



ARM processors are the leading choice of smartphone and tablets. For Apple, its A5 chip used by the iPad 2 although made by Samsung is based on ARM's Cortex-A9 design. ARM designs are also used by Texas Instruments and Qualcomm to power Android-based devices.

Additionally, Microsoft has announced Windows software will support ARM, along with Intel chipsets. Graphics hardware firm Nvidia also plans to use ARM chips in its Tegra processors.

- La société ARM ne vend pas de processeur physique !

- ARM :

Cortex-A5: Area and Power Efficiency

- The ARM Partnership's tremendous success stems from a fundamental focus on efficiency, area, power and cost
  - Billions of consumer units connected to batteries starting from day 1

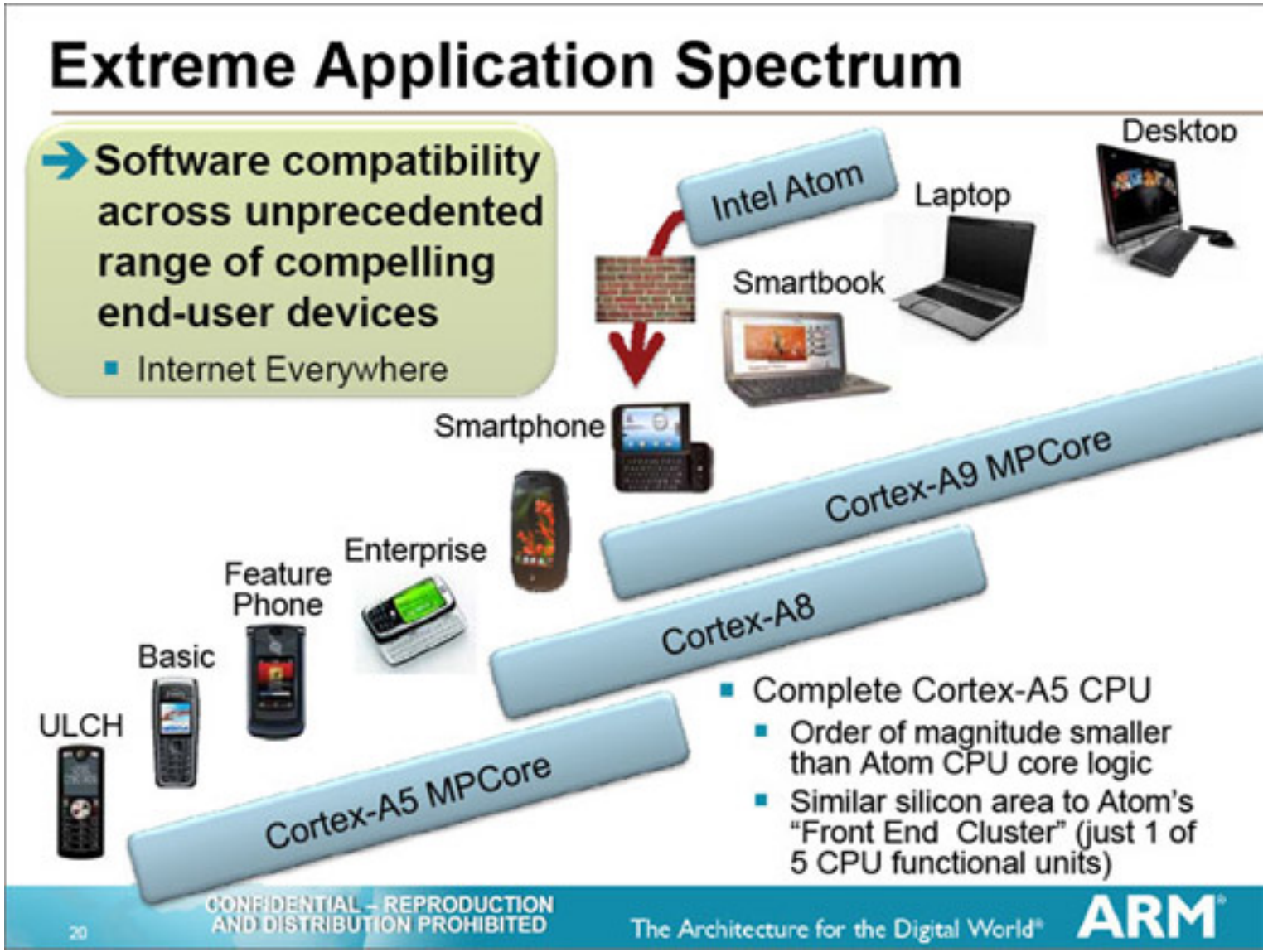
→ Cortex-A5 is an order of magnitude smaller and more power-efficient than Intel Atom's CPU Core

- Less area → less complexity
- Less area → less active power
- Less area → less leakage power
- Less area → less cost

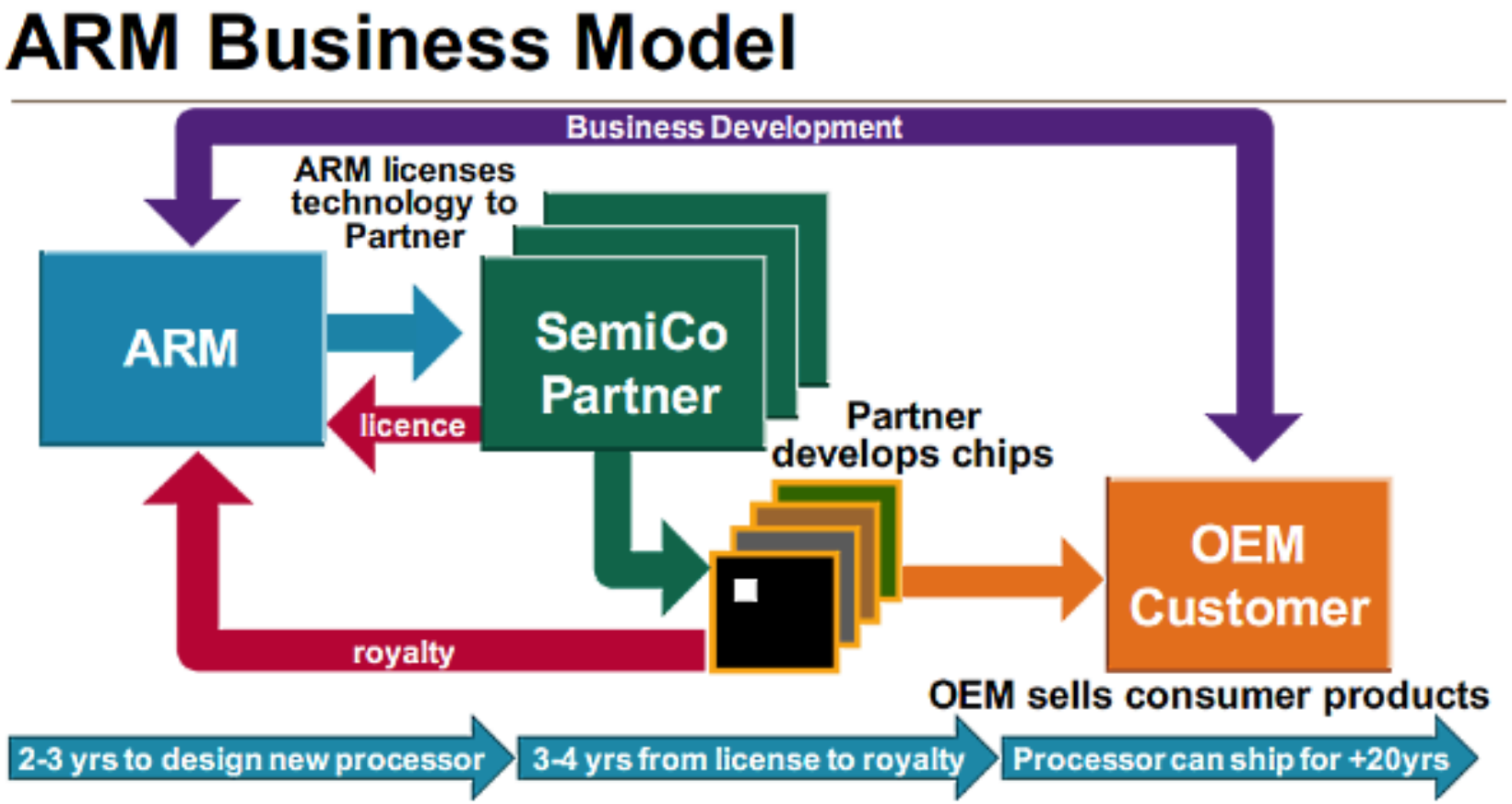
- The Atom CPU would need to be manufactured at **15nm** to match the silicon area of the Cortex-A5 in today's 45nm process

ARM Estimates; Scaled to same manufacturing geometry; Atom die shot is representative

- ARM :


Extreme Application Spectrum
Software compatibility across unprecedented range of compelling end-user devices
Internet Everywhere
Intel Atom
Laptop
Desktop
Smartbook
Smartphone
Enterprise
Feature Phone
Basic
ULCH
Cortex-A9 MPCore
Cortex-A8
Cortex-A5 MPCore
Complete Cortex-A5 CPU
Order of magnitude smaller than Atom CPU core logic
Similar silicon area to Atom's "Front End Cluster" (just 1 of 5 CPU functional units)
20
CONFIDENTIAL – REPRODUCTION AND DISTRIBUTION PROHIBITED
The Architecture for the Digital World®
ARM

La société ARM a en effet fait le choix de ne pas fabriquer de processeurs. Elle se contente de concevoir des architectures (ARMv6, ARMv7, ARMv8…) et diverses microarchitectures de référence implémentant ces architectures (les fameux Cortex), puis vend des licences sur ces architectures. Les fabricants des puces ARM peuvent alors soit prendre une licence sur l'architecture et concevoir leur propre microarchitecture, comme le font par exemple Apple et Qualcomm (et bientôt, nVidia, Samsung…), soit prendre une licence sur une microarchitecture de référence et l'intégrer avec d'autres circuits pour former un SoC complet. Pour ce faire, ils peuvent d'ailleurs également prendre des licences sur d'autres circuits conçus par ARM, notamment les GPU Mali.



I. Notion de numération binaire (le monde du numérique et de l'analogique)

II. Rapide Historique de l'évolution des ordinateurs

III. Le modèle Von Neuman et représentation hiérarchique

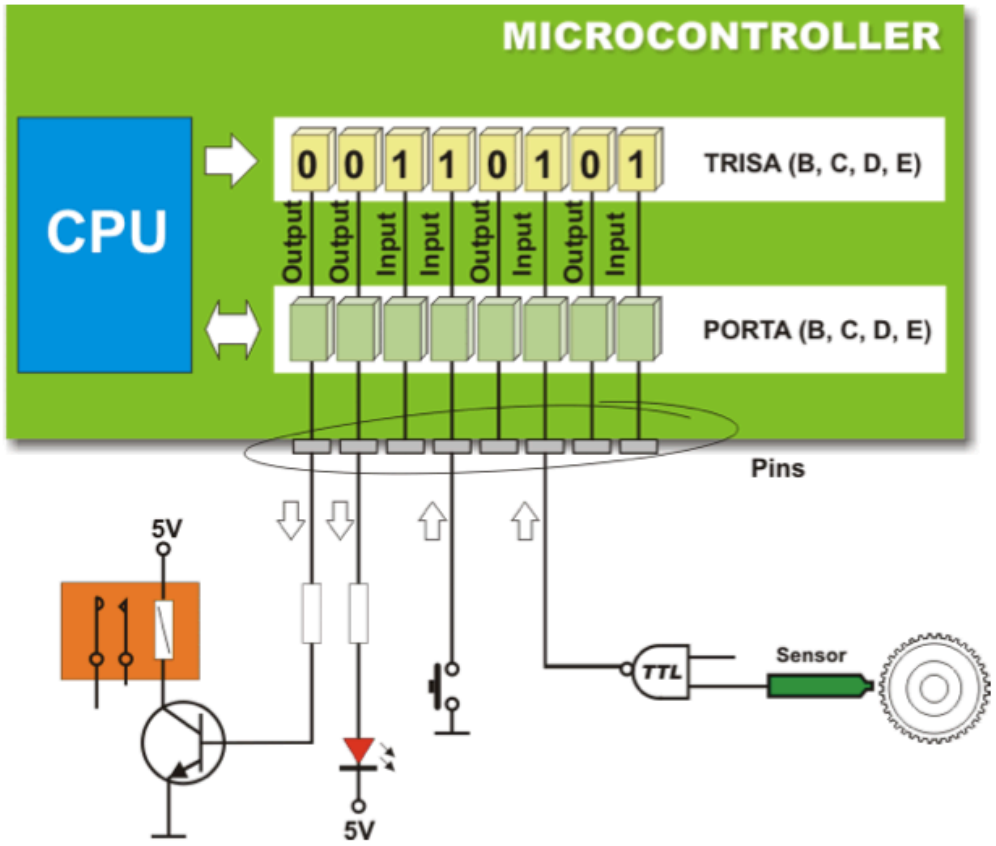
IV. Les différents composants du système et leur caractéristiques

a. Les mémoires

b. Le processeur :

1. modélisation primaire
2. L'horloge
3. Jeu d'instruction
4. **Les interruptions**
5. **Les I/Os multifonctions (compléments)**

- **Cas du PIC : comment placer une patte en Input ou Output ?**
- **Comment placer un « 1 » logique sur une patte ?**



PORTA and TRISA register

Vu en TD

I. Notion de numération binaire (le monde du numérique et de l'analogique)

II. Rapide Historique de l'évolution des ordinateurs

III. Le modèle Von Neuman et représentation hiérarchique

IV. Les différents composants du système et leur caractéristiques

a. Les mémoires

b. Le processeur :

1. modélisation primaire
2. L'horloge
3. Jeu d'instruction
4. **Les interruptions**
5. **Les I/Os multifonctions**
6. **ADC (suite)**

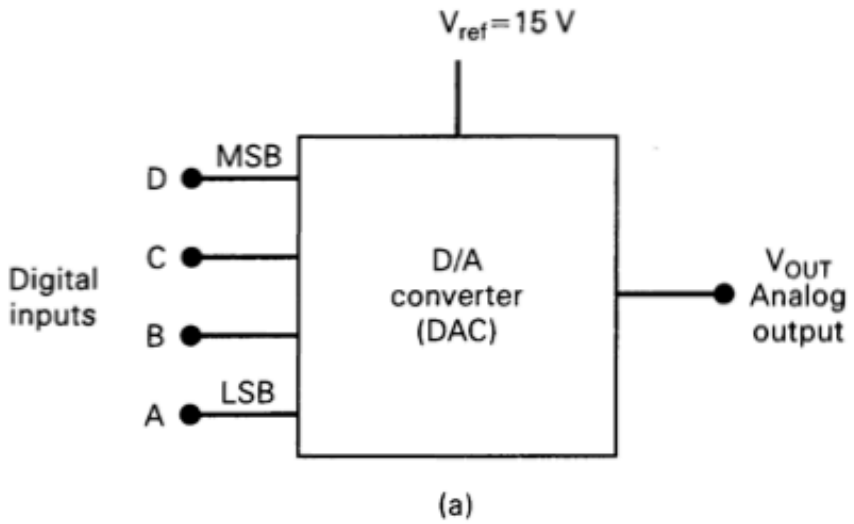
- **Présentation du document : « Le module ADC du PIC 16F88 »**

1. voir diapo suivante présentation ADC et DAC, approximation succ.
2. Fin de la lecture du document « Le module ADC »
3. Exemple utilisation ADC – langage C – PIC (voir diapo suivante)

Le module ADC du PIC 16F88

- 1- Caractéristiques du convertisseur ADC
- 2- Mise en oeuvre
  - **Etape 1 : Configuration**
    - 1-1- Choix des canaux d'entrées
    - 1-2- Choix des tensions de référence
    - 1-3- Choix du format du résultat de la conversion
    - 1-4- Choix de la fréquence d'horloge du convertisseur
    - 1-5- Mise en service de l'interruption du convertisseur
  - Etape 2- Mise en service du convertisseur
  - Etape 3- Sélection du canal à échantillonner
  - Etape 4- Attente pendant la phase d'acquisition
  - Etape 5- Lancement de la phase de conversion
  - Etape 6- Attente de la fin de la phase de conversion

- Principe d'un exemple de **CNA - DAC**



(a)

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

(b)

$$\textbf{analog output} = K \times \textbf{digital input}$$

$$V_{OUT} = (1\ \text{V}) \times \text{digital input}$$

- Principe d'un exemple de CNA DAC : exercice commenté 1

A five-bit DAC has a current output. For a digital input of 10100, an output current of 10 mA is produced. What will I_{OUT} be for a digital input of 11101?

Solution

The digital input 10100_2 is equal to decimal 20. Since $I_{OUT} = 10$ mA for this case, the proportionality factor must be 0.5 mA. Thus, we can find I_{OUT} for any digital input such as $11101_2 = 29_{10}$ as follows:

$$\begin{aligned} I_{OUT} &= (0.5 \text{ mA}) \times 29 \\ &= 14.5 \text{ mA} \end{aligned}$$

Remember, the proportionality factor, K, varies from one DAC to another and depends on the reference voltage.

- Principe d'un exemple de CNA DAC : exercice commenté 2

What is the largest value of output voltage from an eight-bit DAC that produces 1.0 V for a digital input of 00110010?

Solution

$$00110010_2 = 50_{10}$$
$$1.0\ \text{V} = K \times 50$$

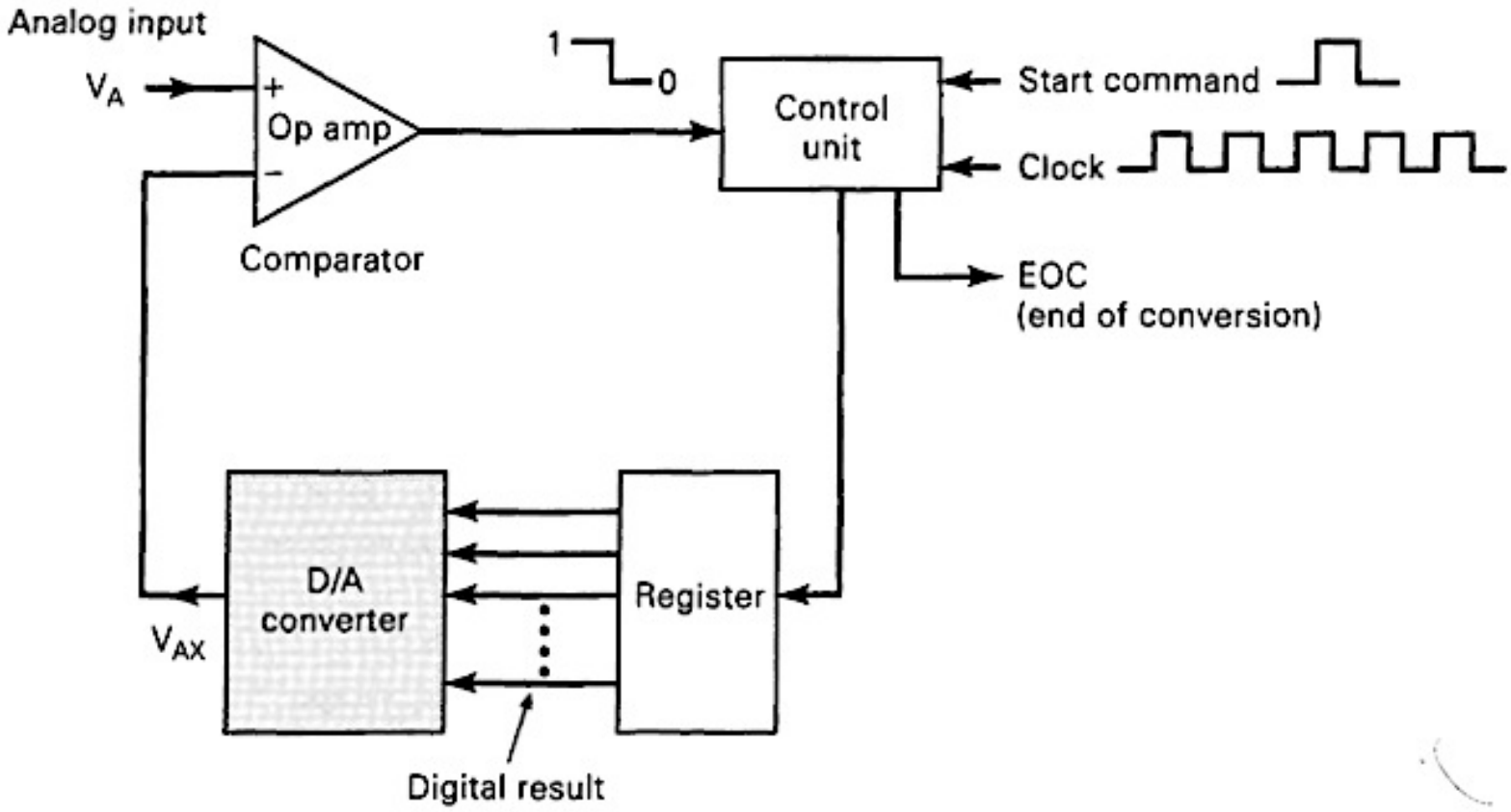
Therefore,

$$K = 20\ \text{mV}$$

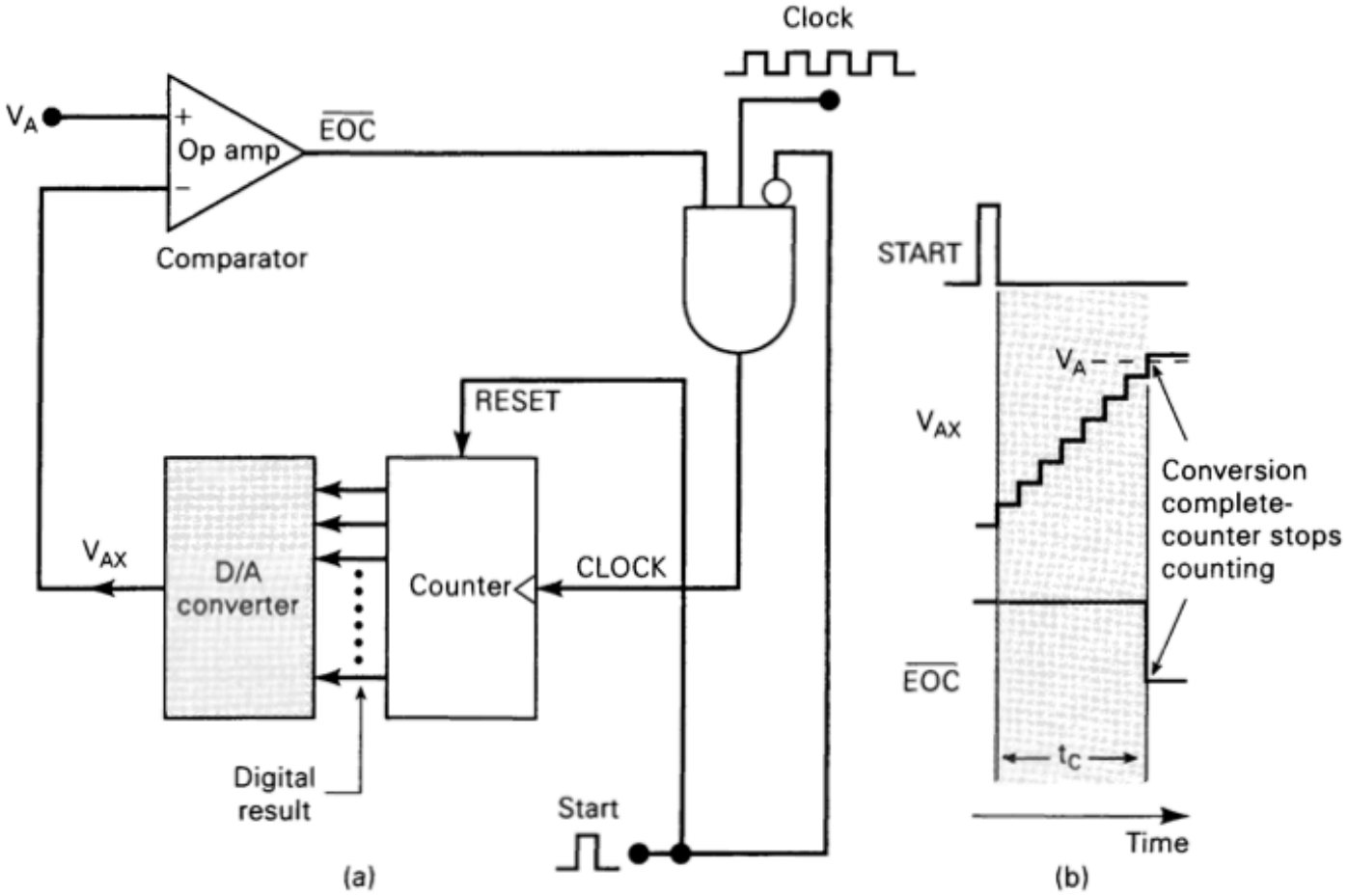
The largest output will occur for an input of $11111111_2 = 255_{10}$.

$$V_{OUT}(\text{max}) = 20\ \text{mV} \times 255$$
$$= 5.10\ \text{V}$$

- Principe d'un exemple de **CAN - ADC**



- Principe d'un exemple de CAN ADC : exercice commenté



6. This continues until V_{AX} reaches a step that exceeds V_A by an amount equal to or greater than V_T (typically 10 to 100 μV). At this point, $\overline{EOC}$ will go LOW and inhibit the flow of pulses into the counter, and the counter will stop counting.

- Assume the following values for the ADC of Figure 10-13: clock frequency = 1 MHz; V_T = 0.1 mV; DAC has F.S. output = 10.23 V and a 10-bit input. Determine the following values.

(a) The digital equivalent obtained for V_A = 3.728 V

(b) The conversion time

(c) The resolution of this converter

Solution

(a) The DAC has a 10-bit input and a 10.23-V F.S. output. Thus, the number of total possible steps is $2^{10} - 1 = 1023$, and so the step size is

$$\frac{10.23\ \text{V}}{1023} = 10\ \text{mV}$$

This means that V_{AX} increases in steps of 10 mV as the counter counts up from 0. Since V_A = 3.728 V and V_T = 0.1 mV, V_{AX} must reach 3.7281 V or more before the comparator switches LOW. This will require

$$\frac{3.7281\ \text{V}}{10\ \text{mV}} = 372.81 = 373\ \text{steps}$$

At the end of the conversion, then, the counter will hold the binary equivalent of 373, which is 0101110101. This is the desired digital equivalent of V_A = 3.728 V, as produced by this ADC.

- ADC **langage C** – PIC: exemple de code

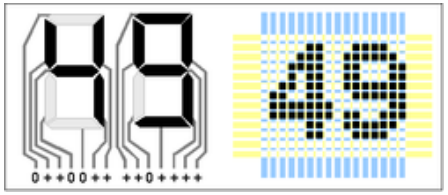
```
#include "16F877A.h"
#device ADC=8                                          //8-bit conversion

#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_D0, rcv=PIN_D1)    //LCD output

void main() //*************************************************
{
  int vin0;                                            // Input variable

  setup_adc(ADC_CLOCK_INTERNAL);                       // ADC clock
  setup_adc_ports(ALL_ANALOG);                         // Input combination
  set_adc_channel(0);                                  // Select RA0

  for(;;)
  {    delay_ms(500);
       vin0=read_adc();                                //Get input byte
       vin0=(vin0/32)+0x30;                            //Convert to ASCII

       putc(254); putc(1); delay_ms(10);               // Clear screen
       printf("Input="); putc(vin0);                   // Display input
  }
}
```

V. Microcontrôleur

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels, porte ET/OU, bascule) ?

- Besoin d'un circuit évolutif et donc programmable

- Réalisation où sont à réaliser des opérations évoluées faisant intervenir des boucles « for…. », des décrémentations de variables, des opérations numériques avancées…..**sinon** combinatoire ou séquentiel
- Besoin d'un circuit de petite taille avec un condensé de fonctions électroniques : oscillateur, CAN, compteur d'impulsions, gestion de petits afficheurs LCD ou LED, pilotage d'afficheurs 7 segments et de petits clavier de saisie, comparateur tension, réveil sur détection logique
- Besoin de fonctions temps évoluée : mise en veille pendant une certaine durée, réveil automatique, fonction de type chronomètre…



- Budget limité : prix d'un petit µC ≈ 1 euros (hors prix des logiciels et des platines de programmation), mais µC ≈ 80 euros aussi !!!

Type	Mémoire	E/S	Botier	Code	Prix	Panier
PIC10F200-I/P	256 x 12	4	DIL08	42670	0,67 € HT 0,80 € TTC	1
PIC10F204-I/P	256 x 12	4	DIL08	42672	0,75 € HT 0,90 € TTC	1
PIC10F206-I/P	512 x 12	4	DIL08	42673	0,75 € HT 0,90 € TTC	1
PIC12C508A	512 x 12	6	DIL08	42100	2,49 € HT 2,99 € TTC	1
PIC12C509	1024 x 12	6	DIL08	42101	1,62 € HT 1,95 € TTC	1
PIC12C671	1024 x 14	6	DIL08	42116	4,08 € HT 4,90 € TTC	1
PIC12CE519A	1024 x 12	6	DIL08	42114	2,46 € HT 2,95 € TTC	1
PIC12F508-I/P	512 x 12	6	DIL08	42674	0,92 € HT 1,10 € TTC	1
PIC12F509-I/P	1024 x 12	6	DIL08	42675	1,00 € HT 1,20 € TTC	1
PIC12F629-I/P	1024 x 14	6	DIL08	16030	1,50 € HT 1,80 € TTC	1
PIC12F675-I/P	1024 x 14	6	DIL08	16031	1,63 € HT 1,95 € TTC	1
PIC16C55RC	512 x 12	20	DIL28	42105	7,21 € HT 8,65 € TTC	1
PIC16C71-04	1024 x 14	20	DIL18	42110	6,79 € HT 8,15 € TTC	1

- Prix ATMEGA

Type	Eeprom	E/S	Boîtier	Code	Prix	Panier
ATMEGA8-16PU	8 kB	23	DIL28	16945	3,58 € HT 4,30 € TTC	1
ATMEGA8L-8PU	8 kB	23	DIL28	16070	3,75 € HT 4,50 € TTC	1
ATMEGA16-16PU	16 kB	32	DIL40	16000	4,83 € HT 5,80 € TTC	1
ATMEGA16L-8PU	16 kB	32	DIL40	16071	6,62 € HT 7,95 € TTC	1
ATMEGA32-16PU	32 kB	32	DIL40	16001	6,17 € HT 7,40 € TTC	1
ATMEGA48PA-PU	4 kB	23	DIL28	16074	3,42 € HT 4,10 € TTC	1
ATMEGA48-20PU	4 kB	23	DIL28	16002	2,75 € HT 3,30 € TTC	1
ATMEGA88-20PU	8 kB	23	DIL28	16003	3,25 € HT 3,90 € TTC	1
ATMEGA88PA-PU	8 kB	23	DIL28	16076	4,08 € HT 4,90 € TTC	1
ATMEGA128-16AU	128 kB	53	TQFP64	16004	11,83 € HT 14,20 € TTC	1
ATMEGA162-16PU	16 kB	35	DIL40	16007	5,79 € HT 6,95 € TTC	1
ATMEGA168-20PU	16 kB	23	DIL28	16008	4,42 € HT 5,30 € TTC	1
ATMEGA328P-PU	32 kB	23	DIL28	16072	3,00 € HT 3,60 € TTC	1
ATMEGA644-20PU	64 kB	32	DIL40	16073	8,25 € HT 9,90 € TTC	1

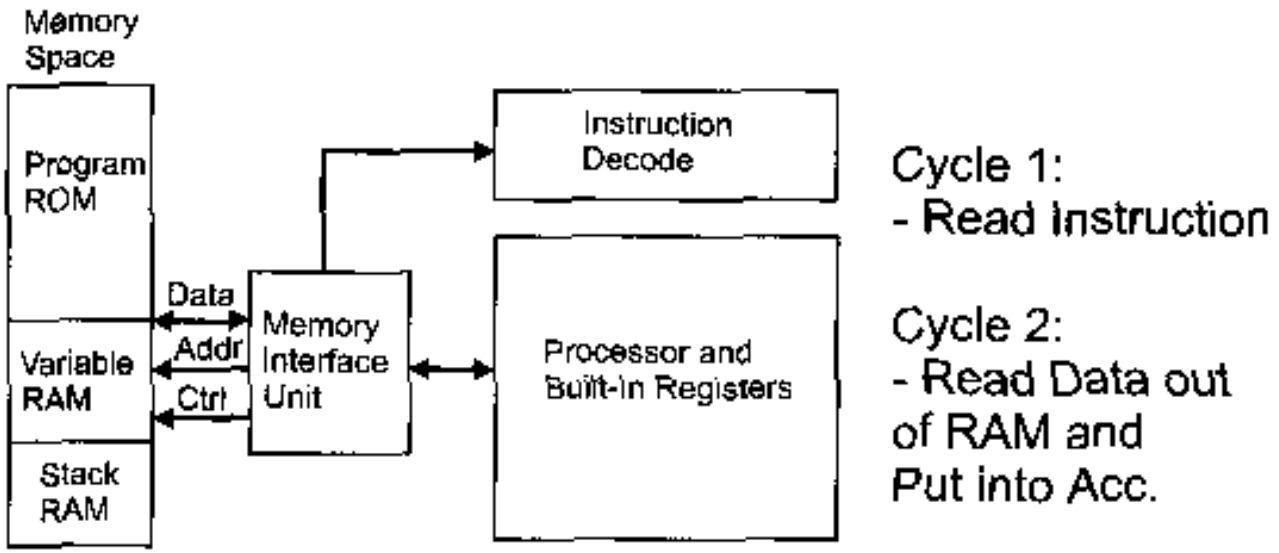
V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC

Un **µP** est un composant **dédié calculs** - Un µP présente une architecture

- **Von Neuman – Princeton :** programme et données sont dans la **même zone mémoire** (la **réalisation** d'un µC sur cette architecture est un peu plus simple), cycles d'instruction séparées.

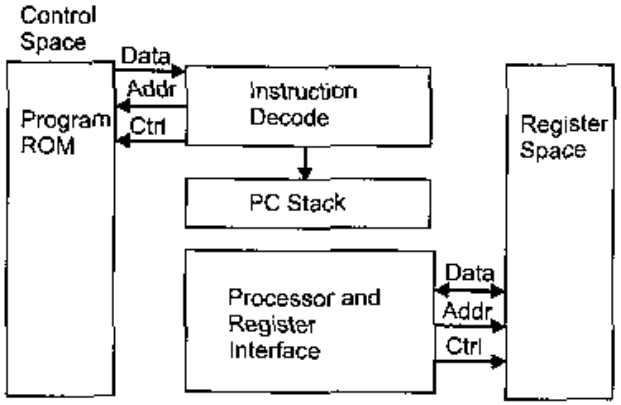
Rappels



"move Acc, Reg" in Princeton Arch.

Un µP présente une architecture

- **Harvard** (**dissociation** des zones de **stockage** données et programme) : complique le design de ces µP mais il y a des avantages : « pipelining » (permet de gagner en temps d'exécution – voir après)

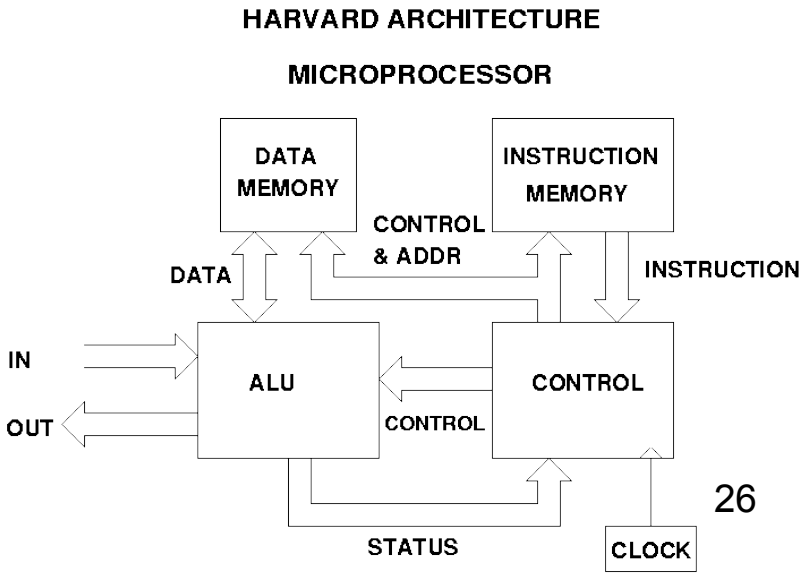


"move Acc, Reg" in Harvard Architect

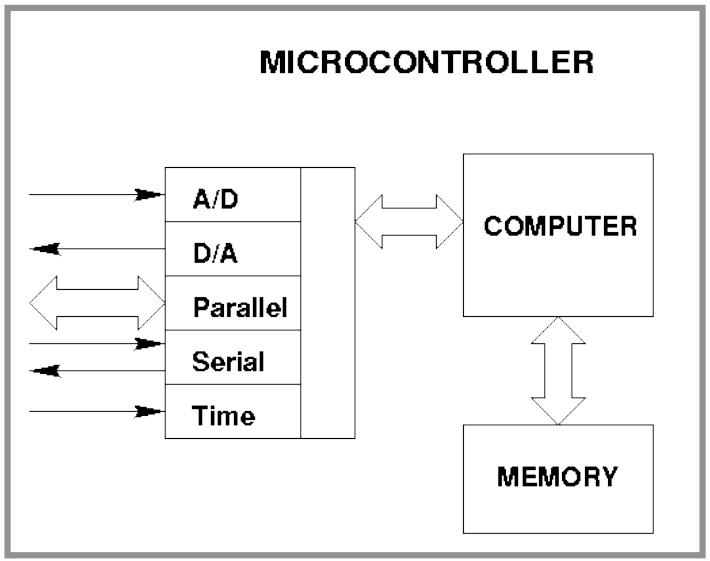
Cycle -1:
- Complete Previous Instruction
- Read the "move Acc, Reg" Instruction

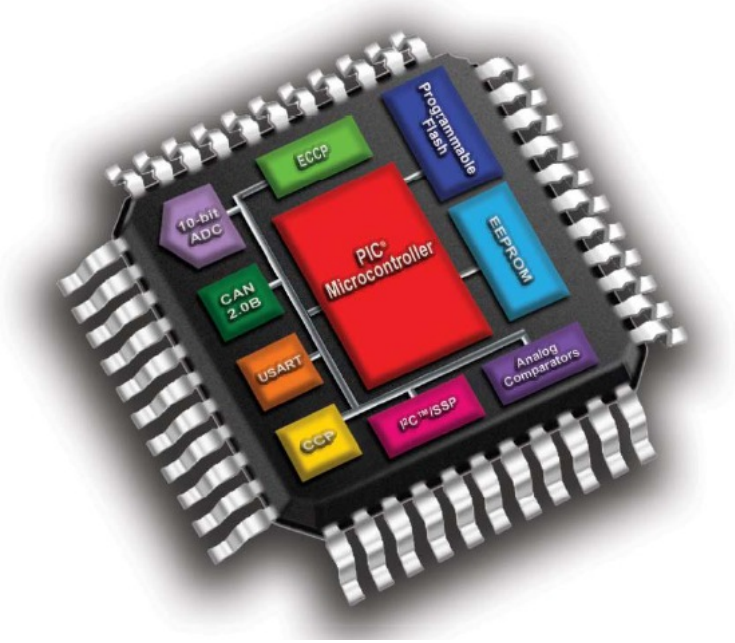
Cycle 1:
- Execute "move Acc, Reg" Instruction
- Read next Instruction

Rappels

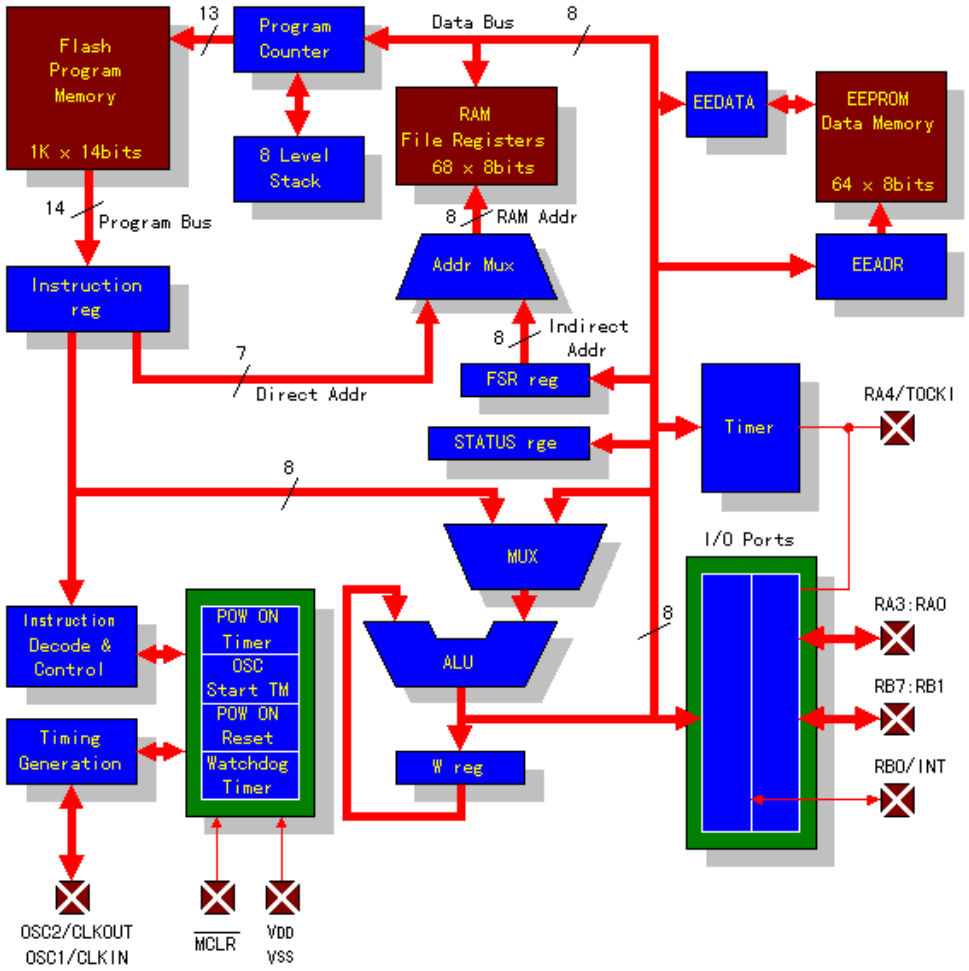


- Un µC est un µP dédié à des interactions avec le monde extérieur
  - CPU : addition, soustraction, décalage…
  - Memory (ROM et RAM, EEPROM, FLASH) : données, programmes…
  - **parallel digital I/O** : commandes numériques
  - Un **module Timer** pour réaliser des opérations en relation avec le temps
  - **serial I/O port** : liaison série pour l'aspect communication
  - **ADC** : pour lire les informations issues des capteurs

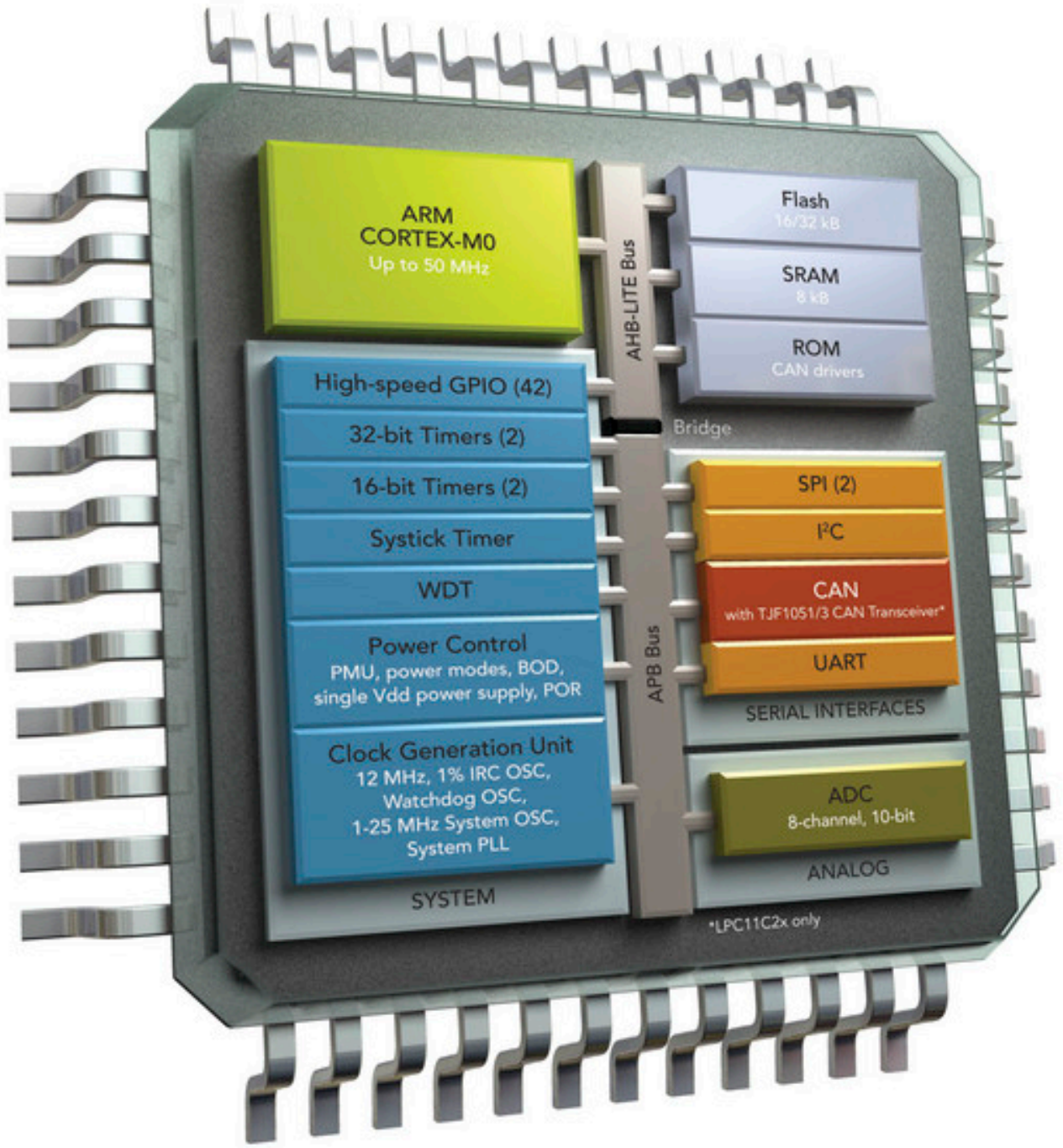




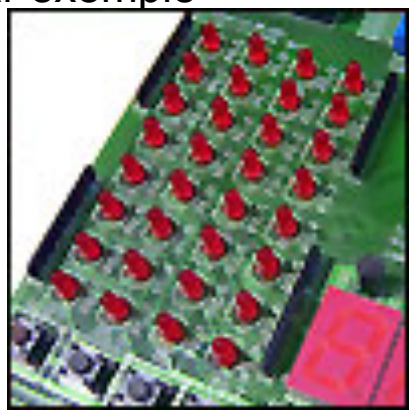
- Un µC est designé pour être opérationnel de façon autonome : possède ce qu'il faut de mémoire, d'organe de calcul….
  - EEPROM, FLASH
  - UAL
  - ADC
  - Bus spéciaux : CAN, I²C….
  - Ports numériques : Ra_i, RB_i

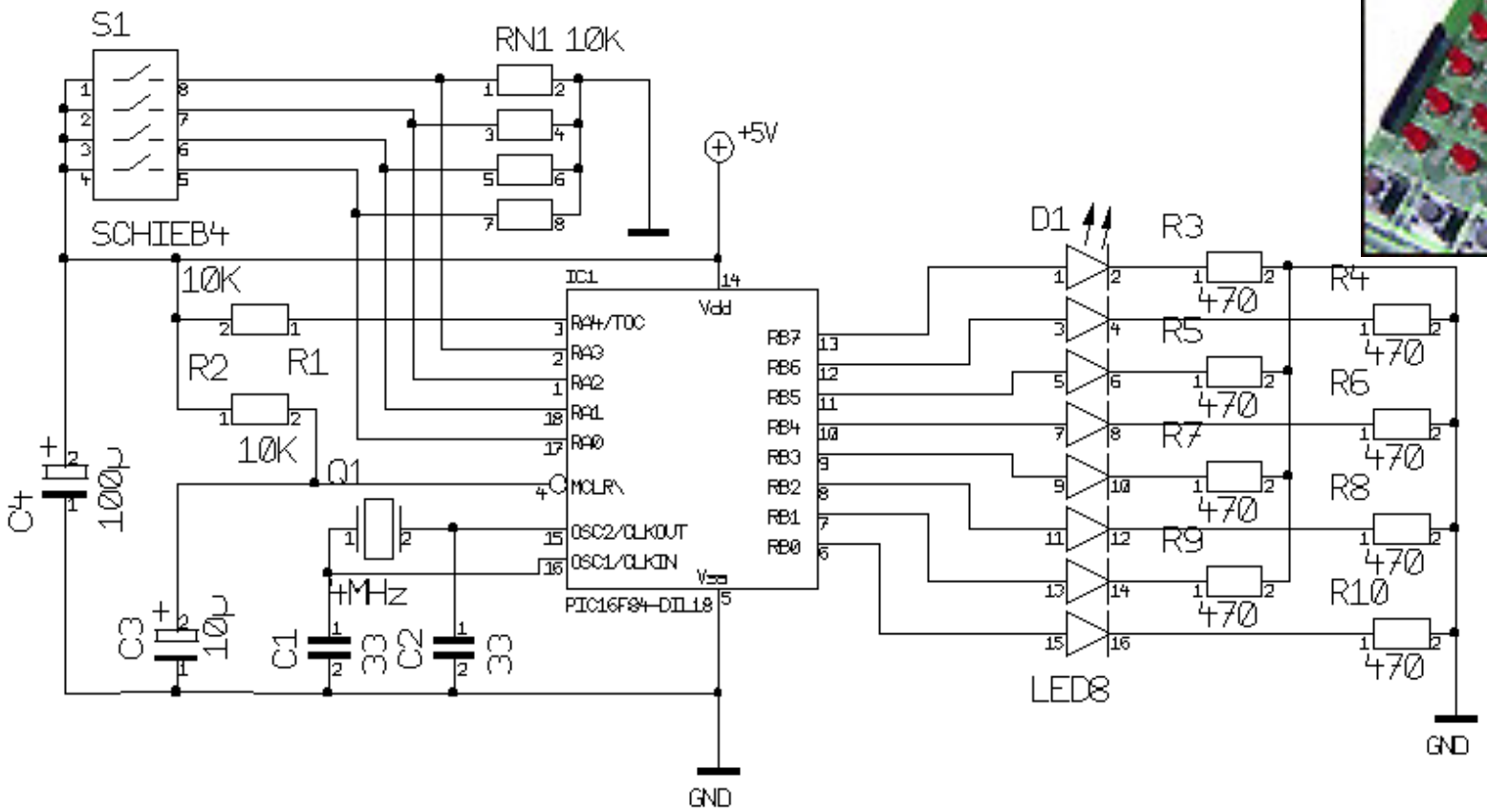


- Autre représentation

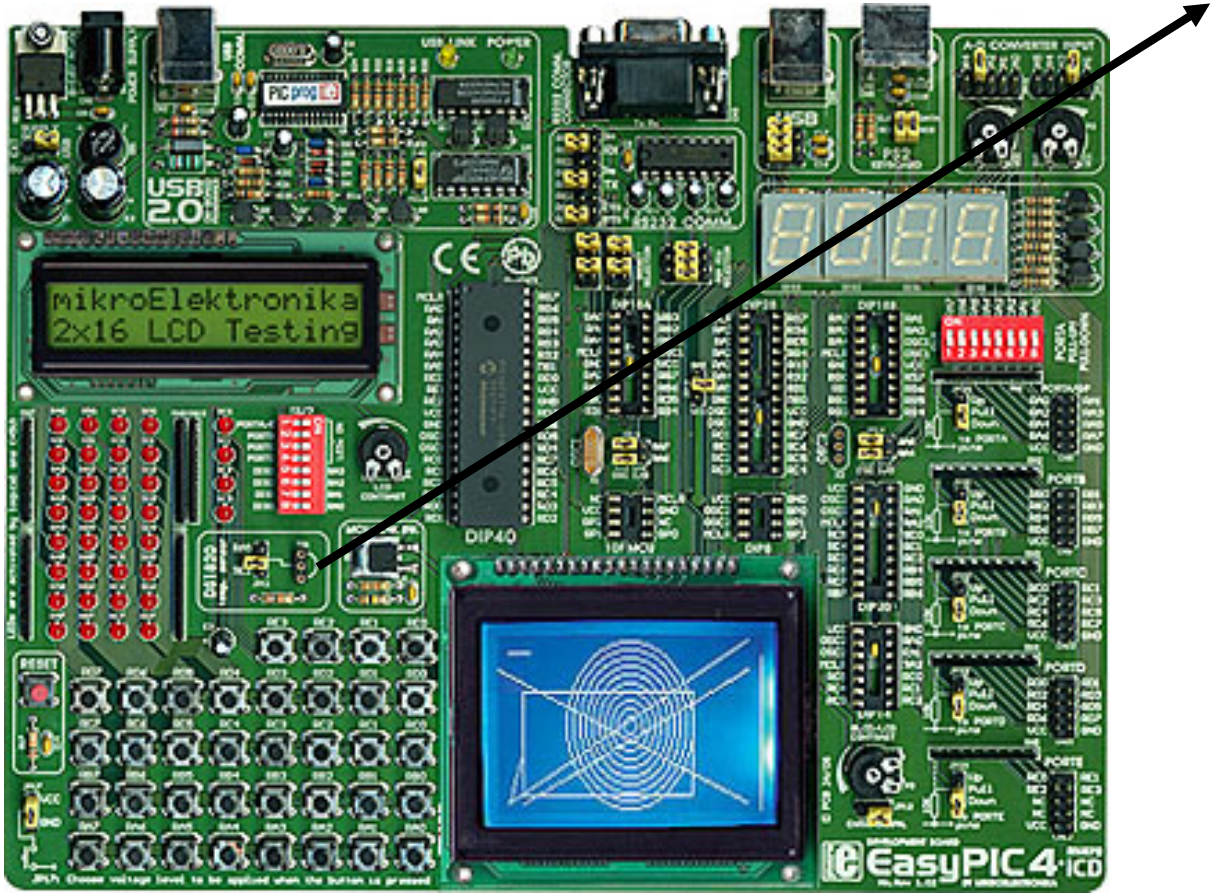


- Un µC
  - Ports numériques : **RBi** utilisés pour le pilotage de **LEDs** par exemple

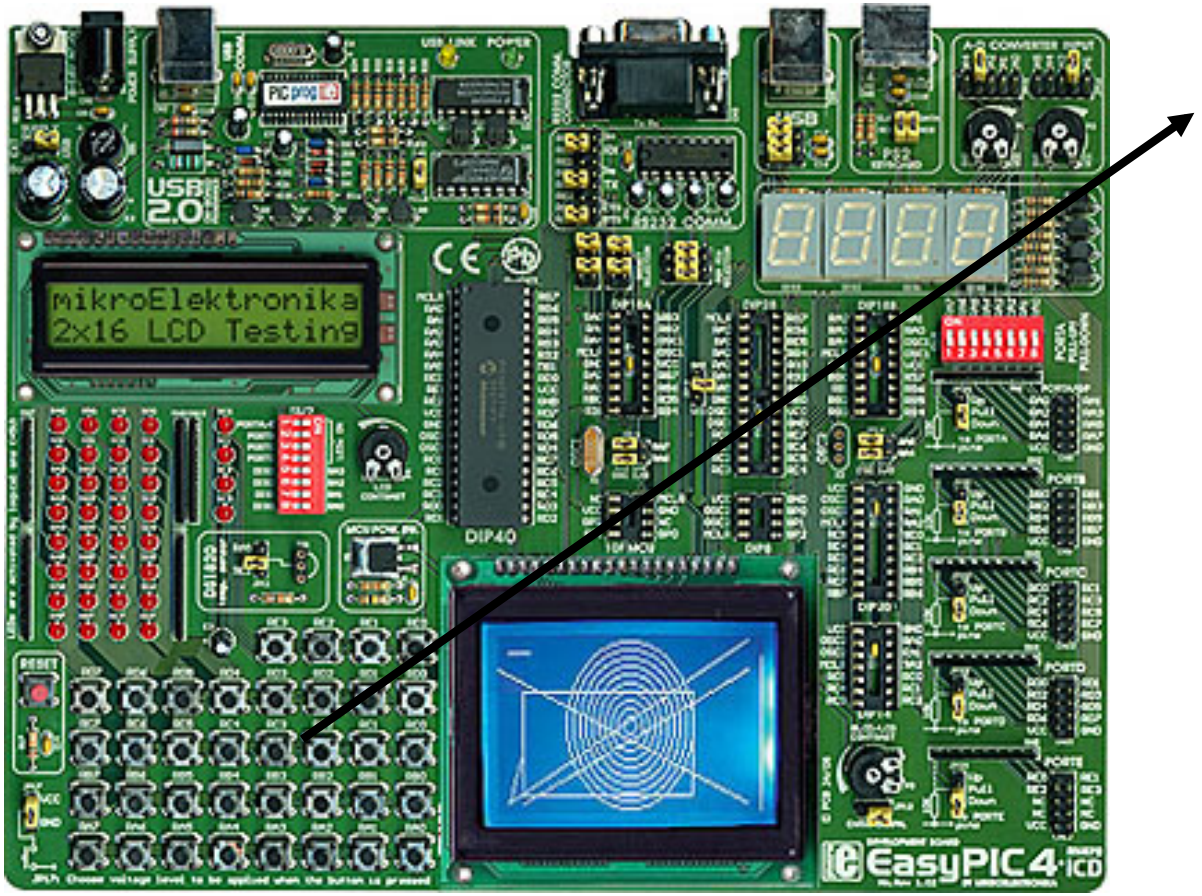


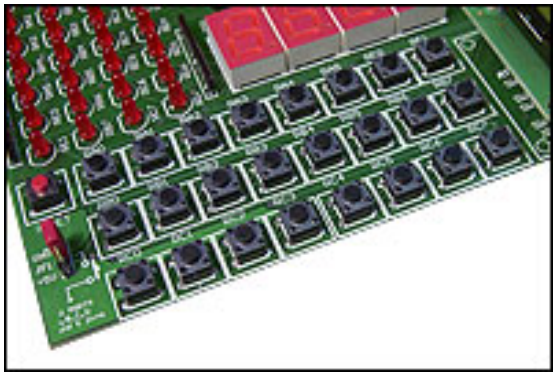


- Un µC
  - **CAN** : pour recueillir des données des capteurs
  - Ex **DS1820** = capteurs de température

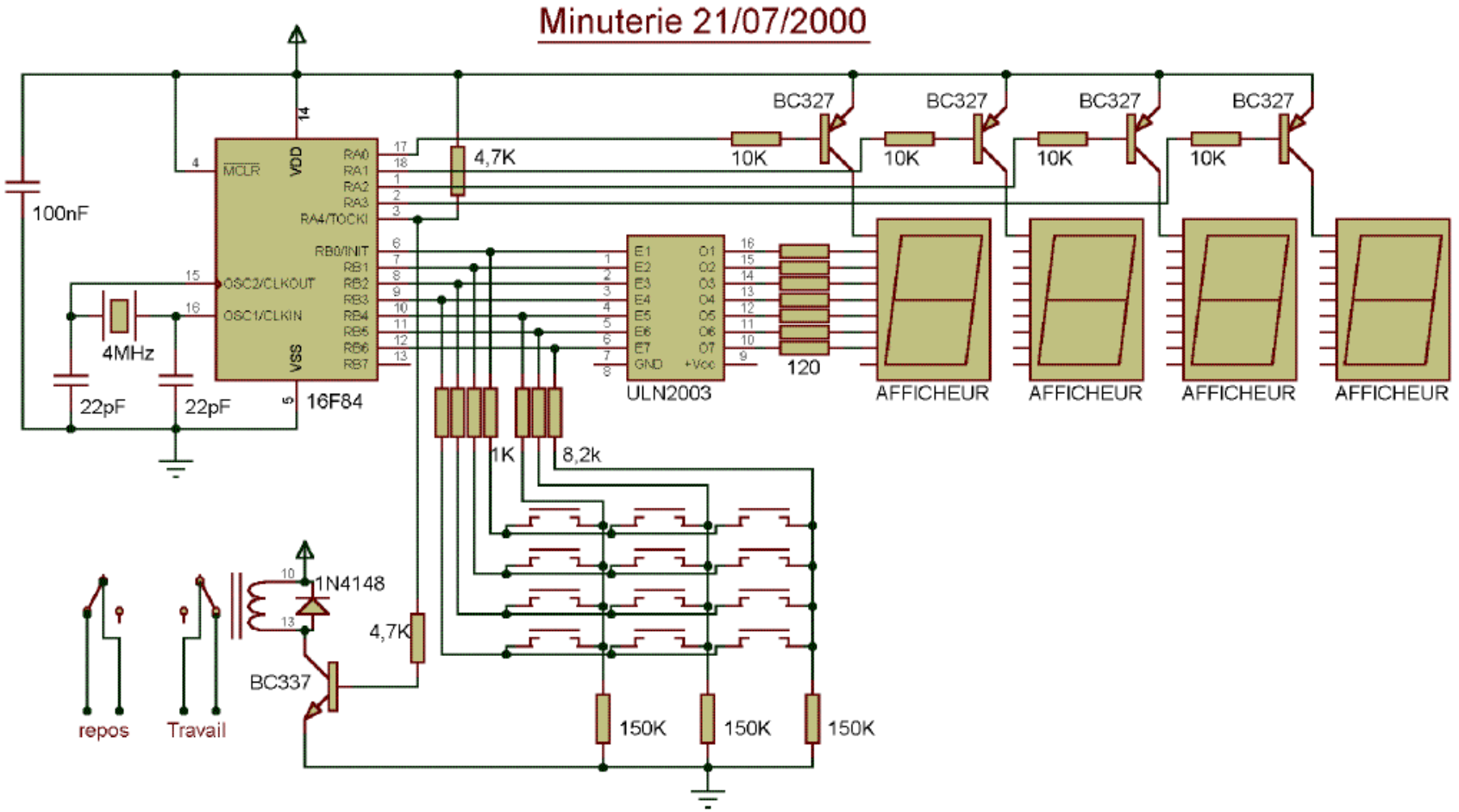


- Un µC
  - **Port numériques** pour détecter le changement d'état **d'interrupteurs**

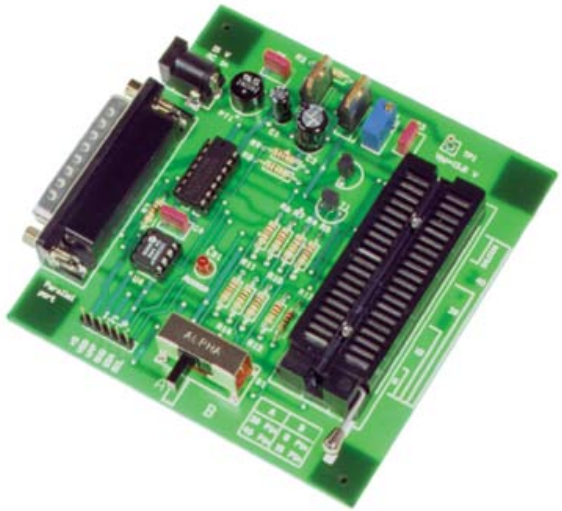




- Un µC
  - Port numériques (Rai ou RBi) pour détecter le **changement d'état d'interrupteurs (clavier, ou interrupteur classique…)**



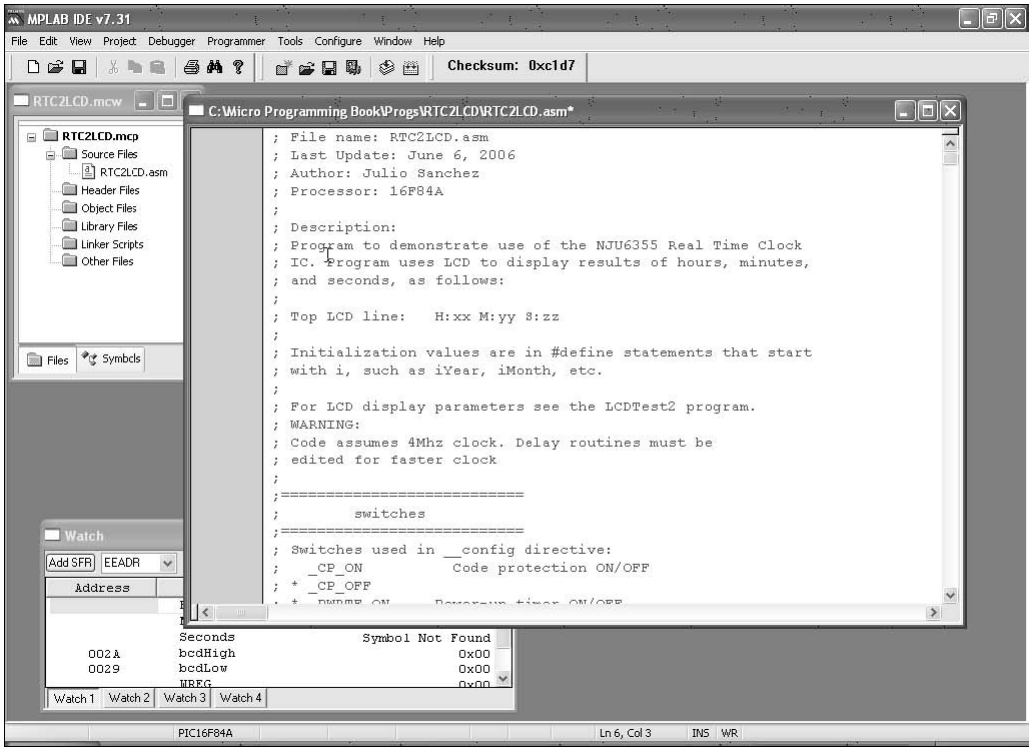
- Un µC
  - **D'interfaces série ou parallèle** pour **communiquer** avec d'autres **systèmes** via des bus (USB, parallèle, série, CAN, I2C, SPI…)



V. Microcontrôleur PIC

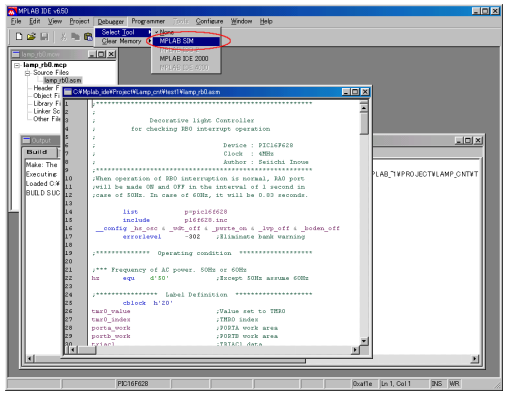
1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?

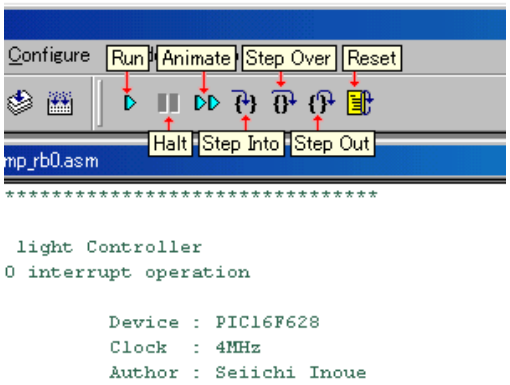
- Choix possible :
- Le code assembleur peut être saisi avec un éditeur de texte « simple » … il est principalement composé de mnémoniques, des paramétrages des fonctions du uC



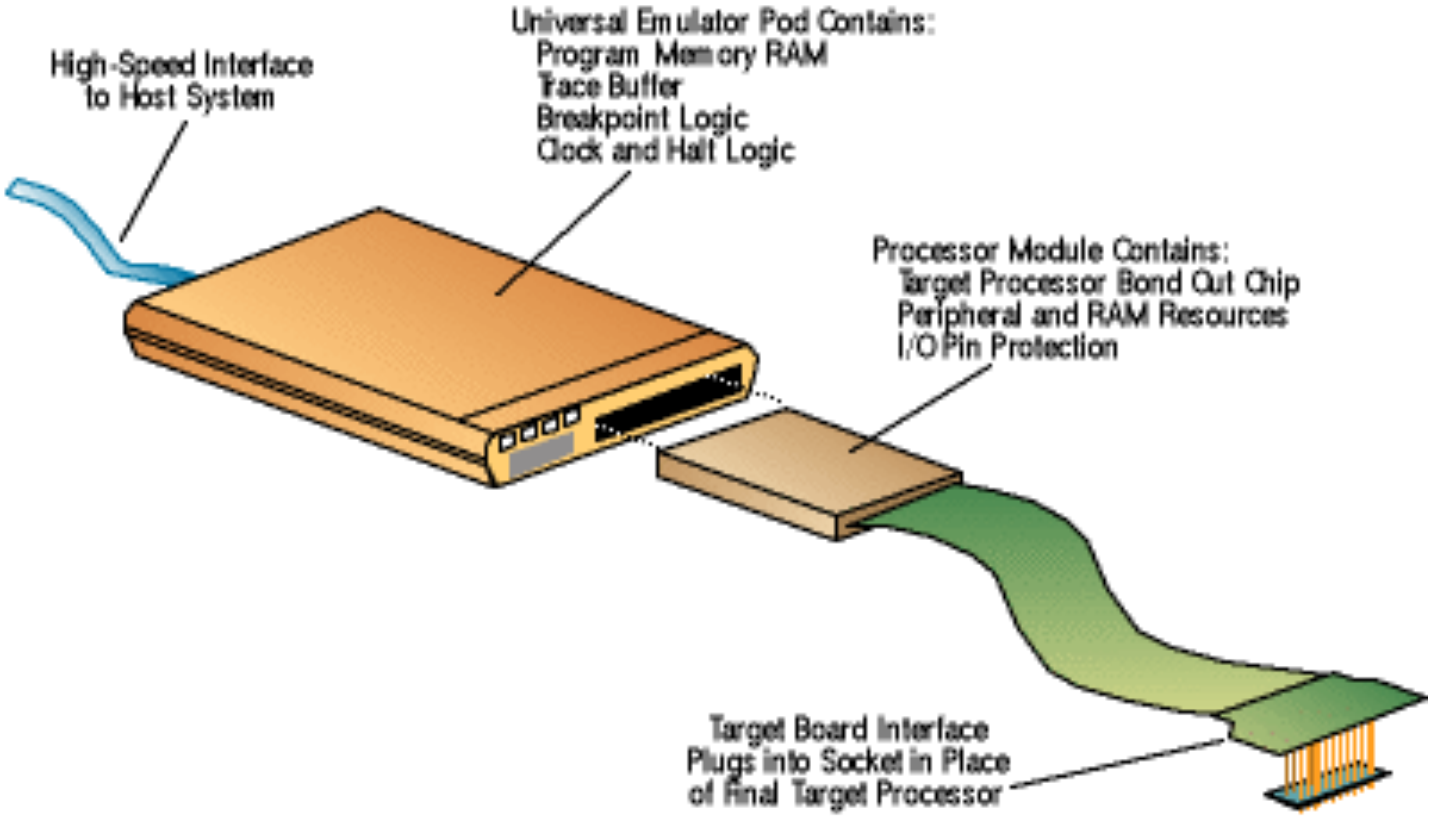
- Choix possible :
- Implanter son code dans le µC directement via l'utilisation d'un **logiciel de « download/programmation »** et voir si ce code fonctionne ! Un peu brut de fonderie mais ….

- Faire une simulation (TPs), la valider, puis implanter le code dans le µC. MPLAB FlowCode

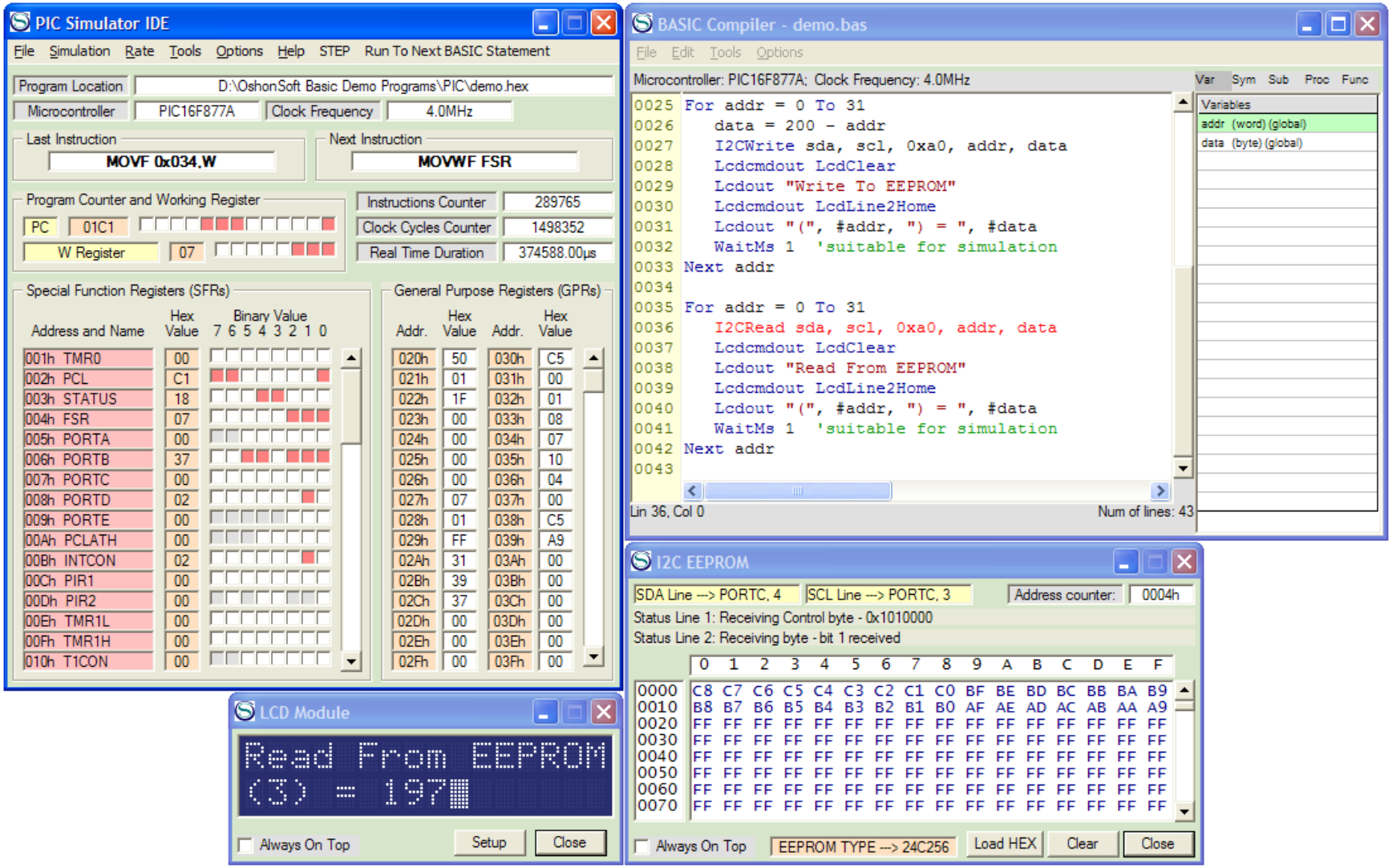




- Choix possible :
- Utiliser un émulateur : plus cher (=remplacer le uC réel par une sonde pilotable par logiciel)



- Choix possible : simulateur software + hardware

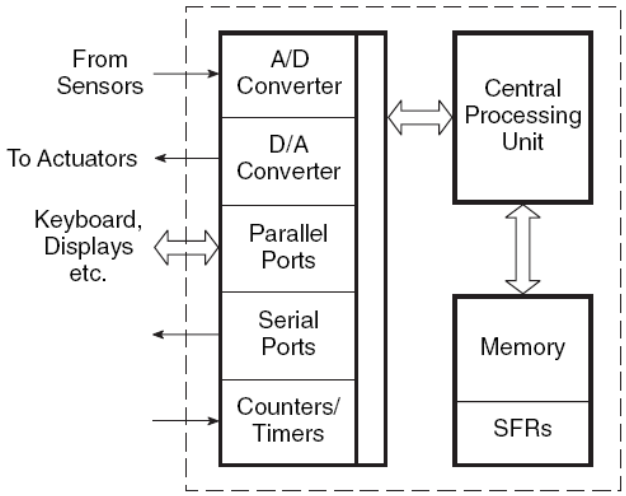


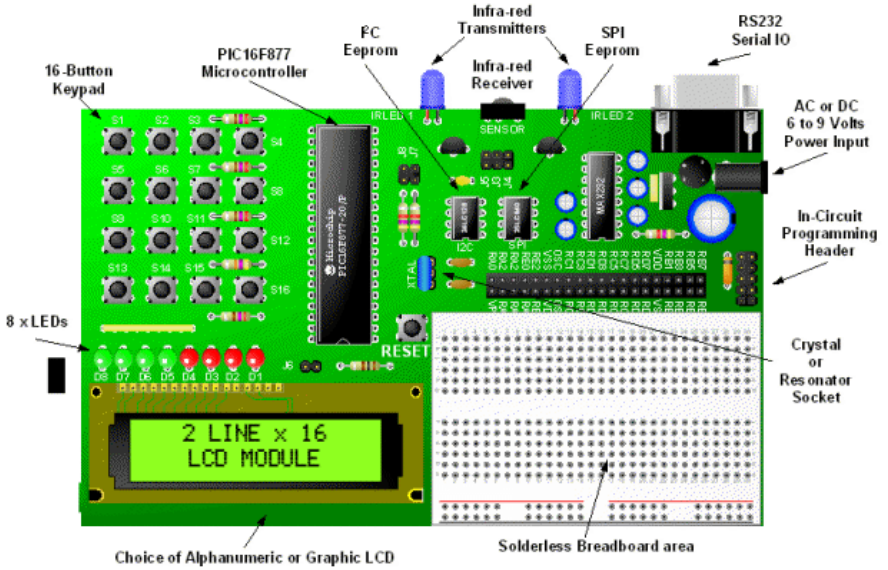
V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?
4. Questions d'avant projet

Questions essentielles à se poser au début d'un projet utilisant un microcontrôleur

- Est-ce que mon application à besoin (questions non exhaustives, mais déjà une bonne approche !!!) :
  - De stocker des données même batterie éteinte (dans un EEPROM) = stockage en mémoire morte (ROM)?
  - Dois je communiquer par liaison série ou parallèle avec d'autres composants ?
  - De combien de ports d'entrée ai-je besoin (dois-je tenir compte d'un interrupteur que peut déclencher un opérateur, pour exécuter une action spécifique) ? Nombre de sorties ?
  - De quelles fonctions ai-je besoin ? Ai-je besoin d'un CAN ? Timer ?
  - Ai-je besoin de LEDs ? Boutons poussoirs ? Mini clavier ? Afficheur LCD ?
  - Mon alimentation doit elle être sur batterie ou sur secteur ? De quelle tension dispose-t-on ?





- En fonction des réponses ⇨ choix de différents microcontrôleurs :

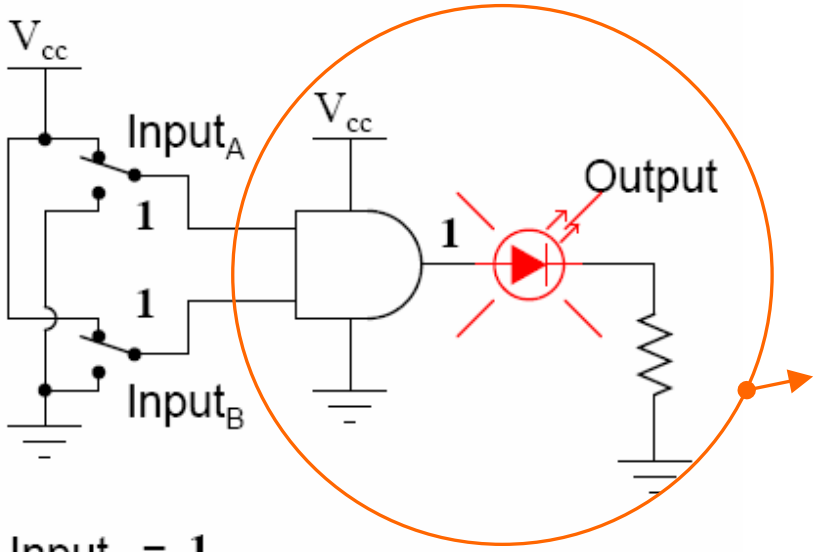
Device number	No. of pins*	Clock speed	Memory (K = Kbytes, i.e. 1024 bytes)	Peripherals/special features
16F84A	18	DC to 20 MHz	1K program memory, 68 bytes RAM, 64 bytes EEPROM	1 8-bit timer 1 5-bit parallel port 1 8-bit parallel port
16LF84A	As above	As above	As above	As above, with extended supply voltage range
16F84A-04	As above	DC to 4 MHz	As above	As above
16F873A	28	DC to 20 MHz	4K program memory 192 bytes RAM, 128 bytes EEPROM	3 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 5 10-bit ADC channels, 2 analog comparators
16F874A	40	DC to 20 MHz	4K program memory 192 bytes RAM, 128 bytes EEPROM	5 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 8 10-bit ADC channels, 2 analog comparators
16F876A	28	DC to 20 MHz	8K program memory 368 bytes RAM, 256 bytes EEPROM	3 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 5 10-bit ADC channels, 2 analog comparators
16F877A	40	DC to 20 MHz	8K program memory 368 bytes RAM, 256 bytes EEPROM	5 parallel ports, 3 counter/timers, 2 capture/compare/PWM modules, 2 serial communication modules, 8 10-bit ADC channels, 2 analog comparators

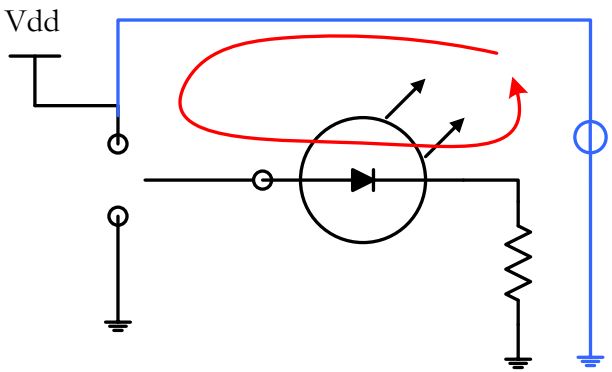
V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?
4. Questions d'avant projet
5. Les interrupteurs contrôlés MOS

Les interrupteurs contrôlés MOS ?? Qu'est-ce ?

- **Pourquoi** avons-nous besoin d'**interrupteurs** électroniques **contrôlés** ?
  - Pour assurer la fermeture/ouverture de certaines mailles électriques afin d'allumer et d'éteindre des composants et des dispositifs électriques (led, relai, moteur, alarme…).
  - Exemple : fermeture d'une boucle de tension (boucle non représentée en général)



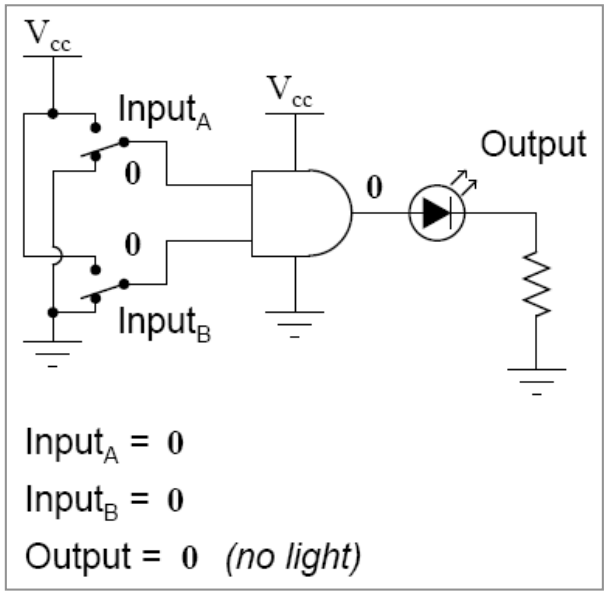


Input$_A$ = 1

Input$_B$ = 1

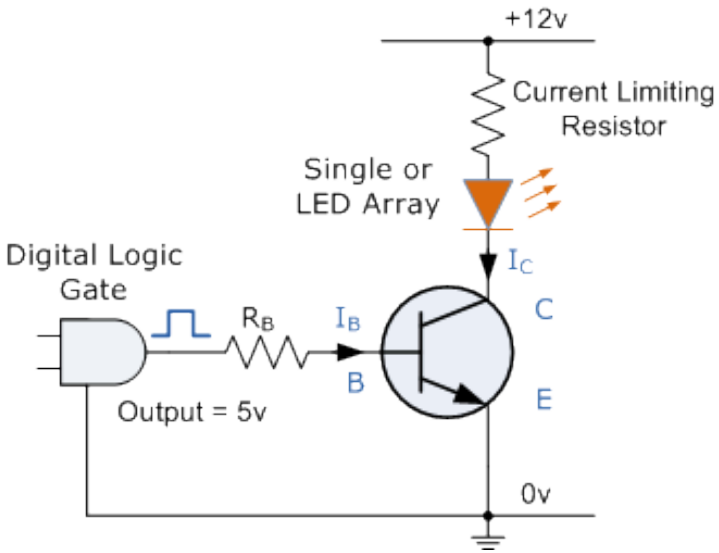
Output = 1 *(light!)*

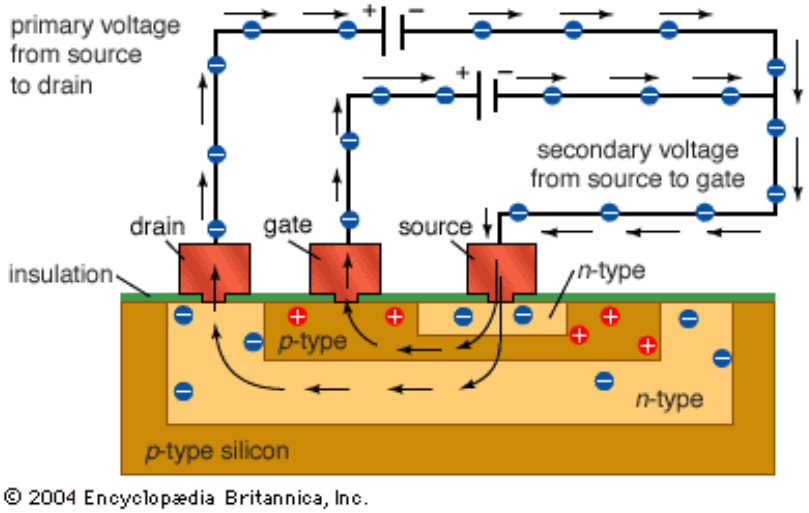
- **Pourquoi** avons-nous besoin d'**interrupteurs** électroniques **contrôlés** ?
  - Dans ce cas le LED reste éteinte.



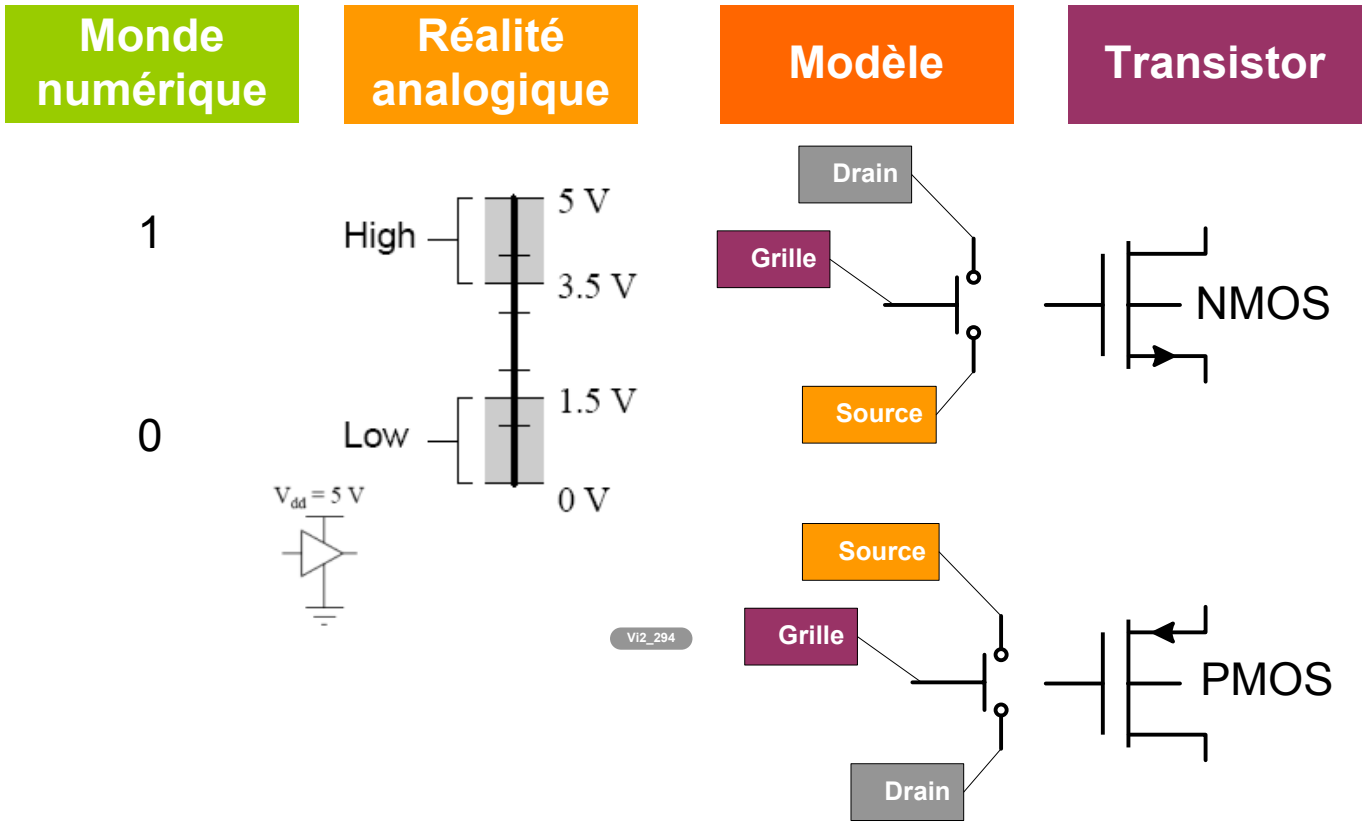
L3

- D'autre part les interrupteurs contrôlés sont à **la base de l'électronique moderne** :
  - Commande = entrée nécessitant une **faible énergie**
  - La grandeur commandée est beaucoup plus grande que la commande (*100, *1000, *10 000)

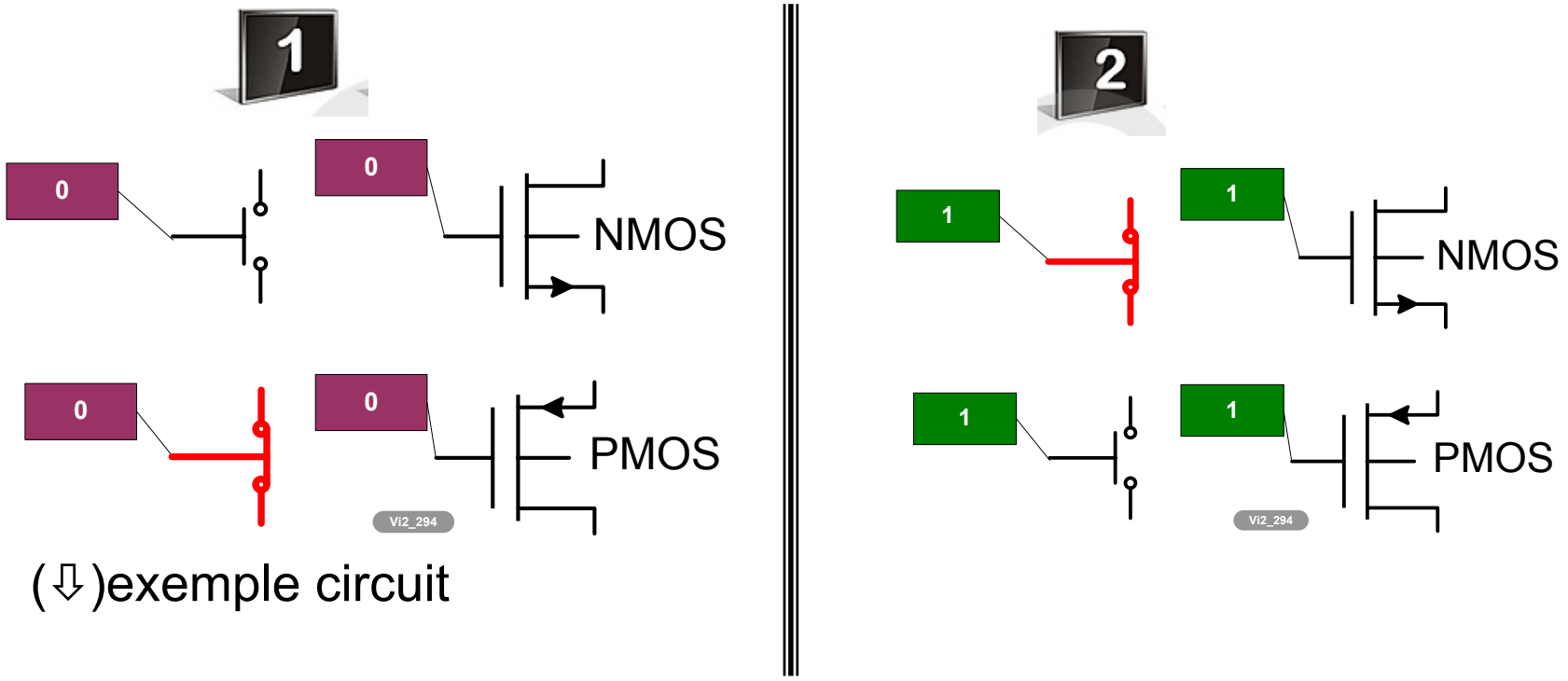




- Fonctionnement d'un transistor MOS en interrupteur :

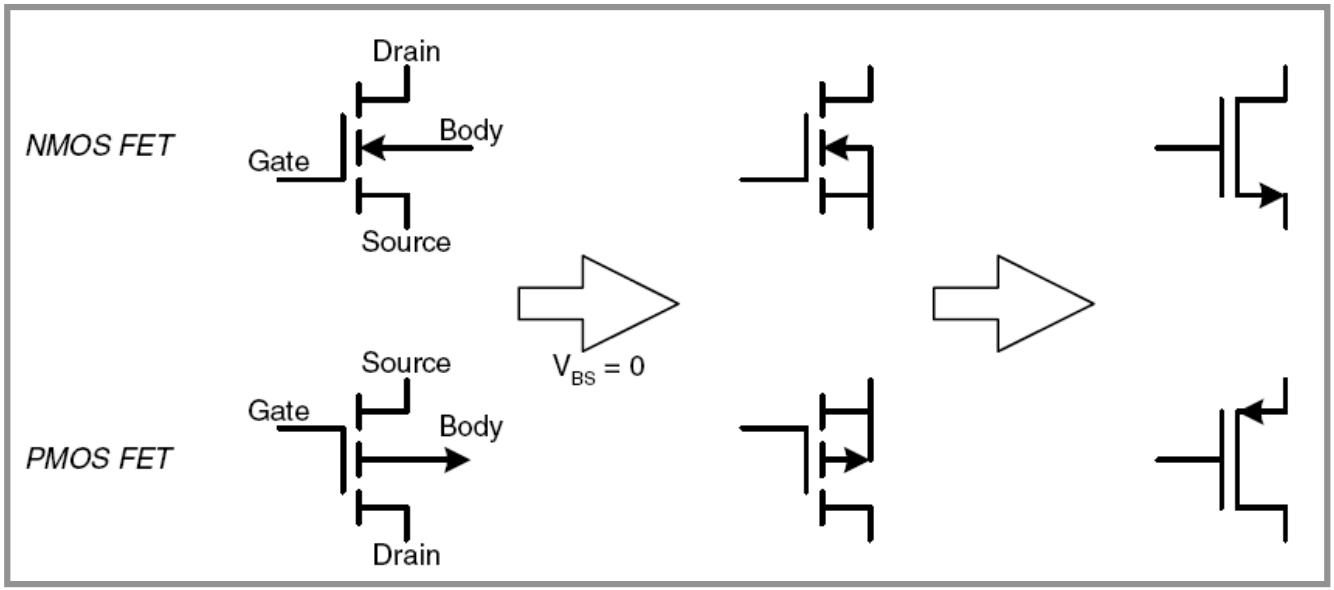


- Fonctionnement d'un transistor MOS en interrupteur :
- Configuration 1 : 0 sur la grille et {drain et source correctement connectés} ⇨ PMOS on, NMOS off
- Configuration 2 : 1 sur la grille et {drain et source correctement connectés} ⇨ PMOS off, NMOS on

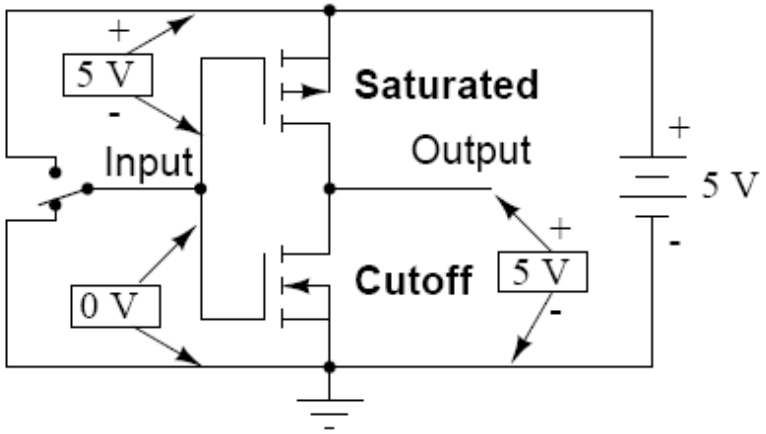


- (⇩)exemple circuit

- Autre symbole de PMOS et de NMOS :

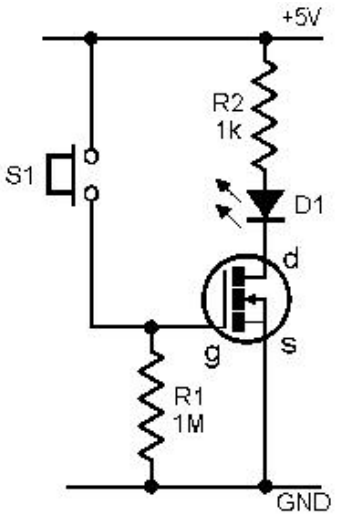


- Exemple de circuit d'application : inverseur CMOS.
- Pourquoi inverseur ?



- Fonctionnement d'un transistor MOS en interrupteur :

1



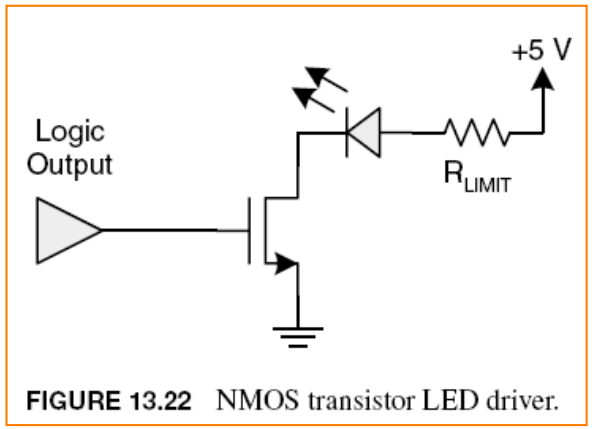
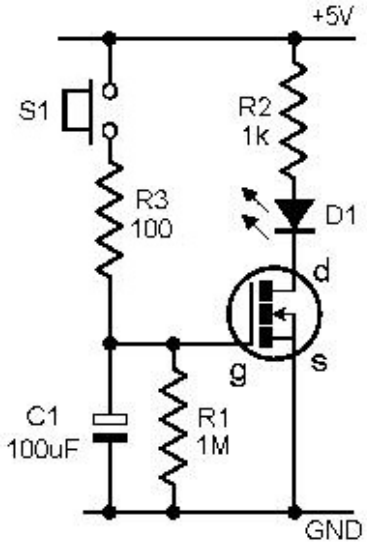


FIGURE 13.22 NMOS transistor LED driver.

2



2 minutes on circuit.....

Press the push button. The LED will stay on when the button is released and remain at full brightness for about a minute and a half then gradually fade away to zero.

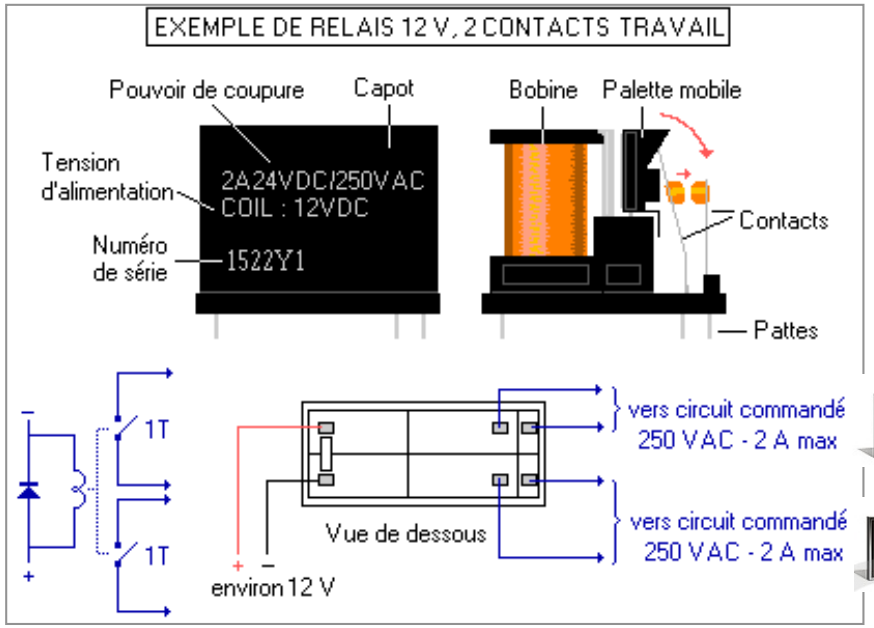
http://brunningsoftware.co.uk/FET.htm

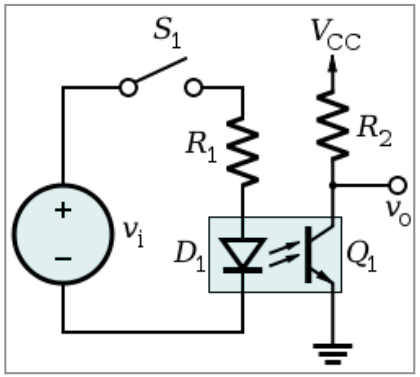
V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?
4. Questions d'avant projet
5. Les interrupteurs contrôlés MOS
6. Autres types d'interrupteurs

Existe-t-il d'autres interrupteurs permettant de fermer des mailles d'un circuit ?

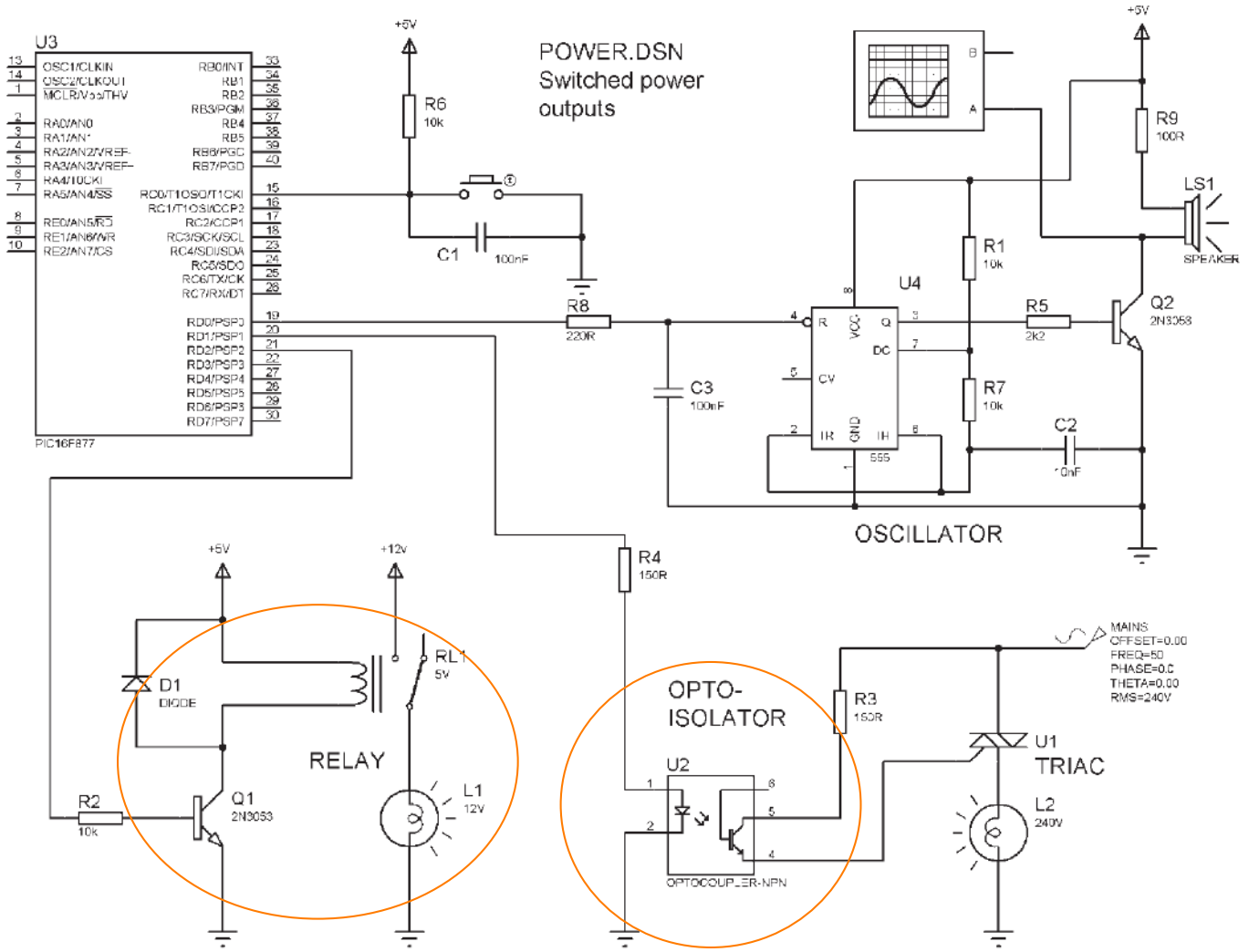
- Les autres types d'interrupteurs…EXTERNES au µC : **relais et optocoupleur** (de « interfacing PIC microcontroller »)
- Nous retrouvons le même principe : commande nécessitant une faible quantité d'énergie et la grandeur commandée est importante en terme de valeur de tension ou de courant.
  - Relais : l'interrupteur est mécanique (solénoïde, fermer 2 sorties en « même temps »)
  - Opto coupleur : l'interrupteur est un phototransistor (que la lumière qui passe = isolement électrique)





Question : peut-on utiliser les 2 secondaires d'un relai en même temps sur une même charge ? Utilité ?

- Les autres types d'interrupteurs : **relais et optocoupleur** (de « interfacing PIC microcontroller »)

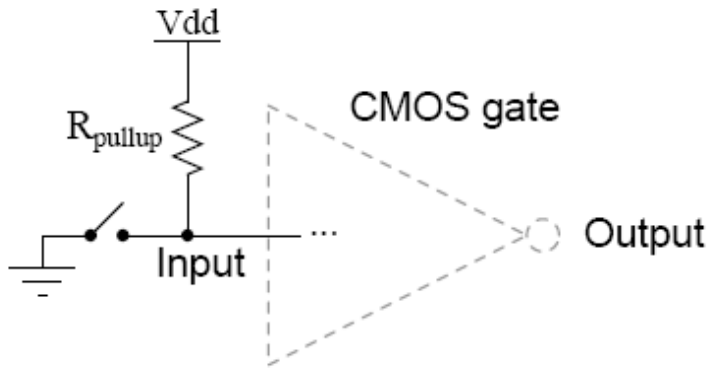


V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?
4. Questions d'avant projet
5. Les interrupteurs contrôlés MOS
6. autres types d'interrupteurs
7. Notion de pull-up pull-down ?

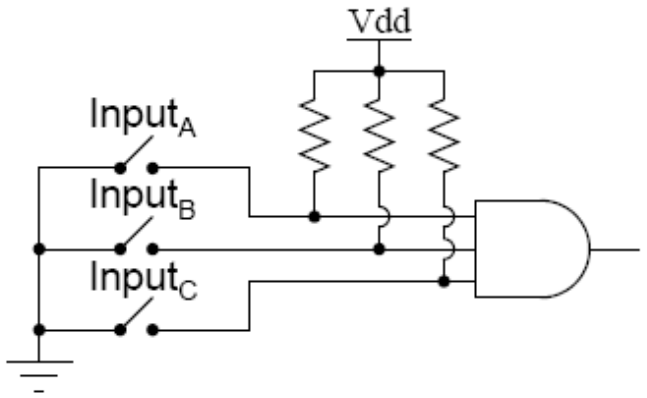
Comment appliquer un 1 ou un 0 sur une entrée du µC ?

- Qu'est-ce que le pullup/down ?



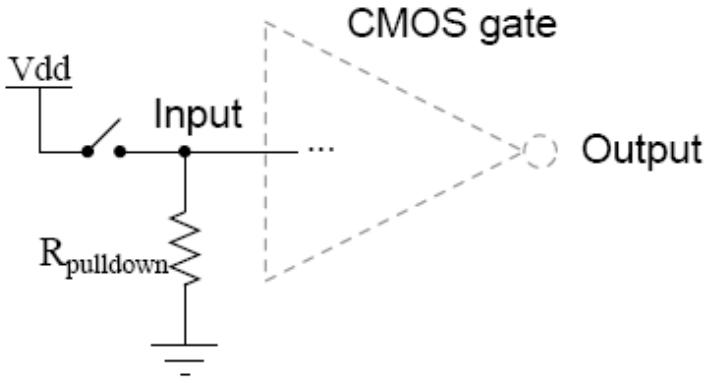
When switch is closed, the gate sees a definite "low" (0) input. When the switch is open, R_{pullup} will provide the connection to Vdd needed to secure a reliable "high" logic level for the CMOS gate input.

Pullup resistors for a 3-input CMOS AND gate



Vocabulaire :
- **Pullup**
- **Pulldown**

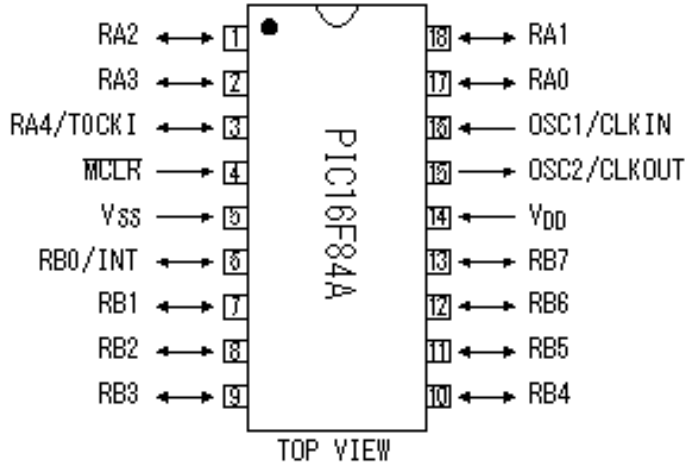
•Activable ou désactivable par programmation



When switch is closed, the gate sees a definite "high" (1) input. When the switch is open, $R_{pulldown}$ will provide the connection to ground needed to secure a reliable "low" logic level for the CMOS gate input.

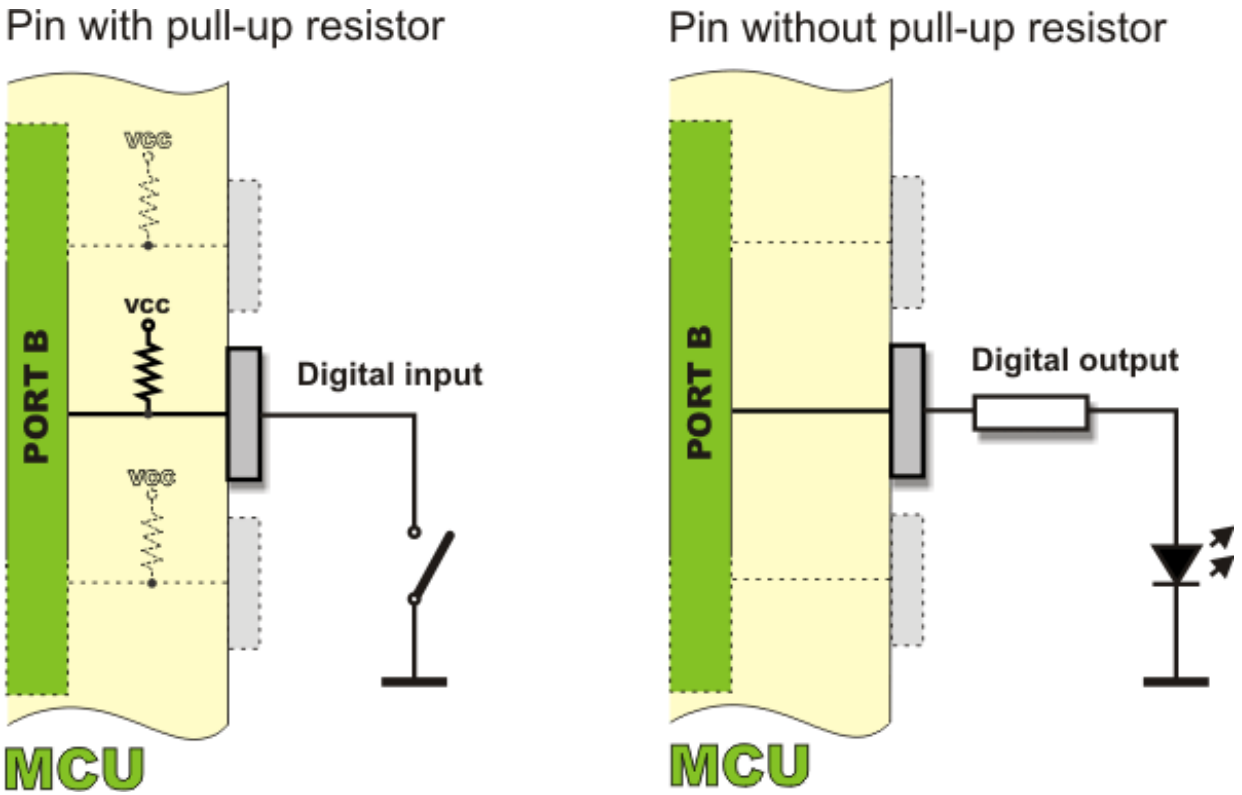
Comment appliquer un 1 ou un 0 sur une entrée du µC ?

- Utilité ?
  - Éviter le phénomène d'antenne associé à une patte d'entrée qui serait laissé flottante = connectée à aucun potentiel fixe
  - Le phénomène d'antenne -> impossible de prévoir la valeur de la patte d'entrée



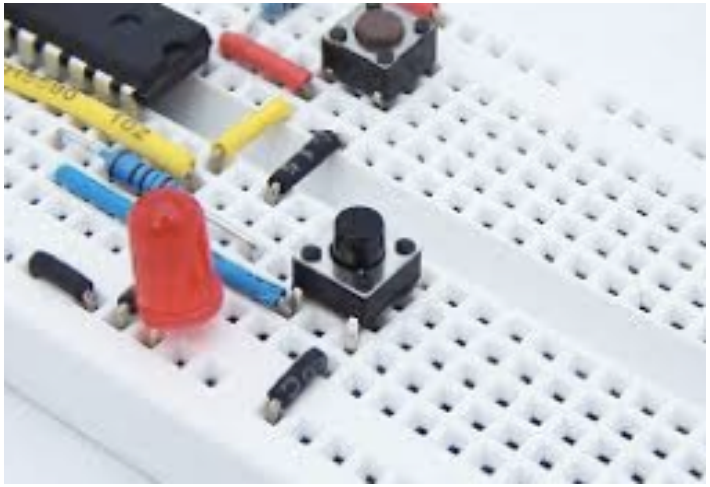
Comment appliquer un 1 ou un 0 sur une entrée du µC ?

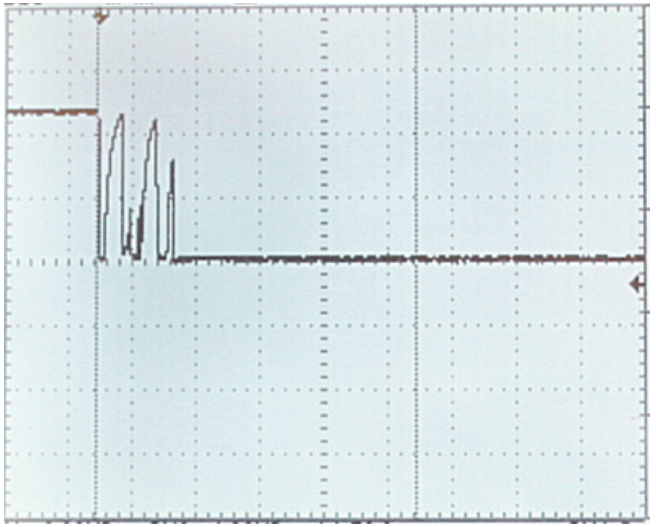
- Pour lire une variable d'entrée (interrupteur), mise en action du pull-up pour éviter des phénomènes aléatoires
- Pour commander une LED …. Inutile de placer le pull-up (le Rpull-up peut même allumer, si elle est activée, la LED, de par sa présence et sa connexion à Vcc)



Comment appliquer un 1 ou un 0 sur une entrée du µC ?

- Notion de rebond d'interrupteur – BOUNCE / DEBOUNCE
- Exemple : fermeture d'un interrupteur

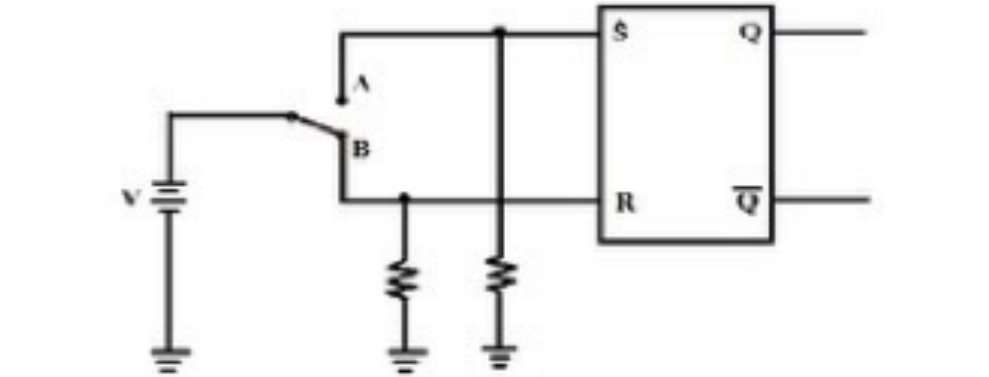




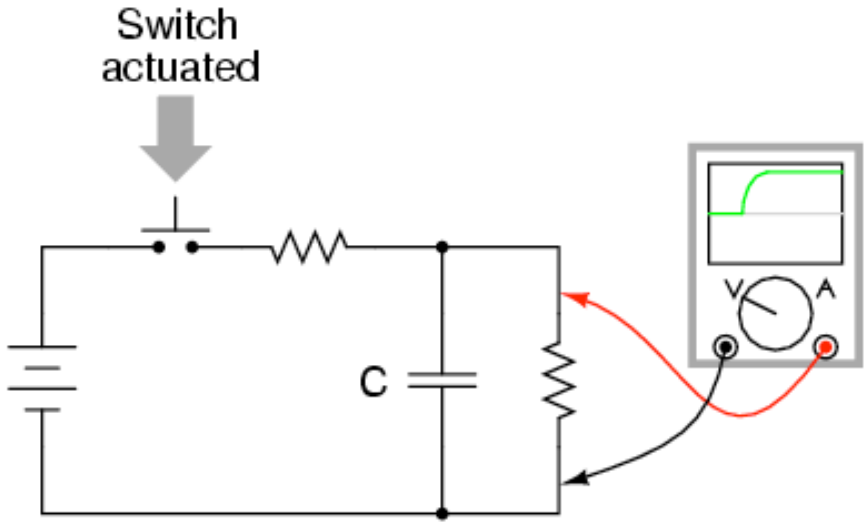
- Est-ce un problème pour les composants numériques ?

Comment appliquer un 1 ou un 0 sur une entrée du µC ?

- Solutions : HARDWARE 1 et HARDWARE 2



Bascule RS

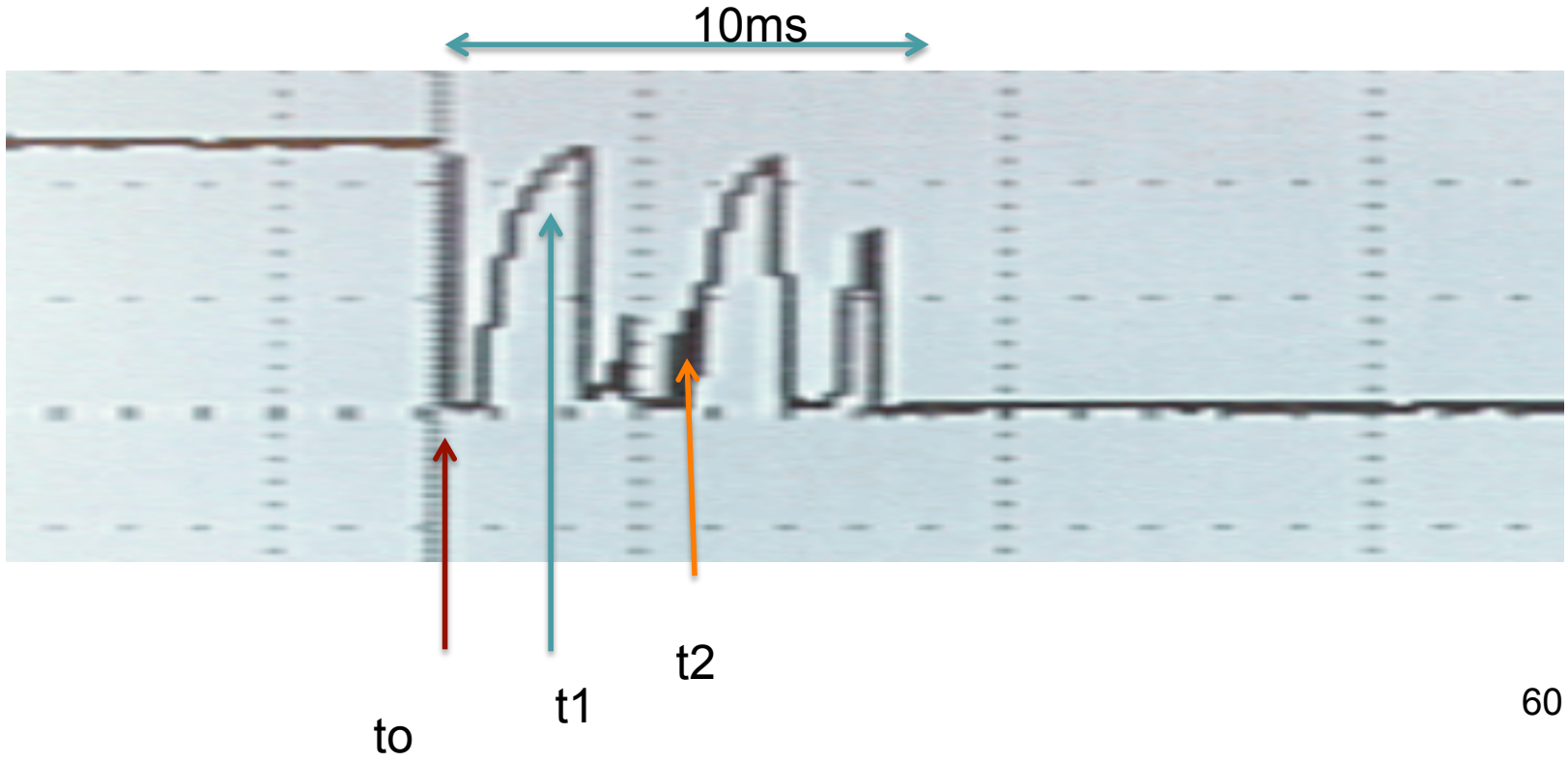


Filtre RC

Comment appliquer un 1 ou un 0 sur une entrée du µC ?

- Solutions : SOFTWARE

- Exemple algorithme : si (valeur logique de l'entrée à to ≠ valeur logique à t1) && (t1-to < 10ms), alors ne pas tenir compte de la valeur de l'entrée et relancer une lecture à t2. Ceci afin de refaire la comparaison précédente



- Solutions : SOFTWARE

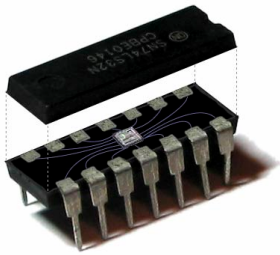
- Exemple code

```
#define MAX_CHECKS 10           // # checks before a switch is debounced
uint8_t Debounced_State;        // Debounced state of the switches
uint8_t State[MAX_CHECKS];      // Array that maintains bounce status
uint8_t Index;                  // Pointer into State

// Service routine called by a timer interrupt
void DebounceSwitch3()
{
    uint8_t i,j;
    State[Index]=RawKeyPressed();
    ++Index;
    j=0xff;
    for(i=0; i<MAX_CHECKS;i++)j=j & State[i];
    Debounced_State= j;
    if(Index>=MAX_CHECKS)Index=0;
}
```

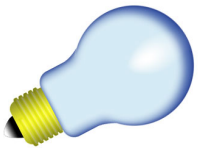
Listing 3: Code that debounces many switches at the same time

V. Microcontrôleur PIC

1. Quand utiliser un µP ou µC au lieu des circuits classiques (combinatoires ou séquentiels)
2. Différence entre µP et µC
3. Comment tester son code assembleur ?
4. Questions d'avant projet
5. Les interrupteurs contrôlés MOS
6. autres types d'interrupteurs



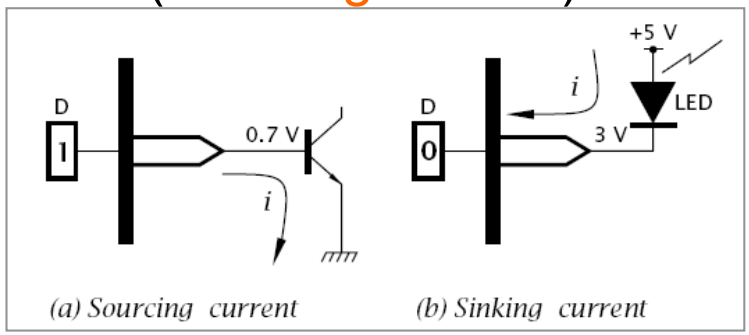
7. Zéro ou un sur une entrée (= 1 patte) du µC ?
8. Limites (min & max) de tensions et de courants pour les I/Os

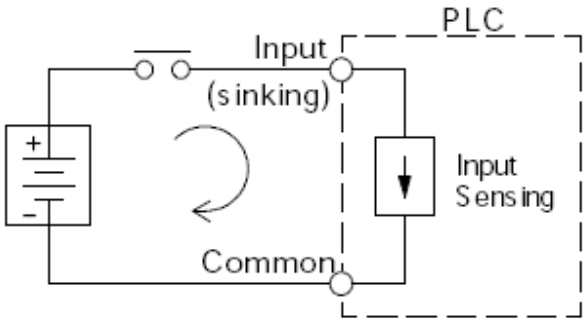


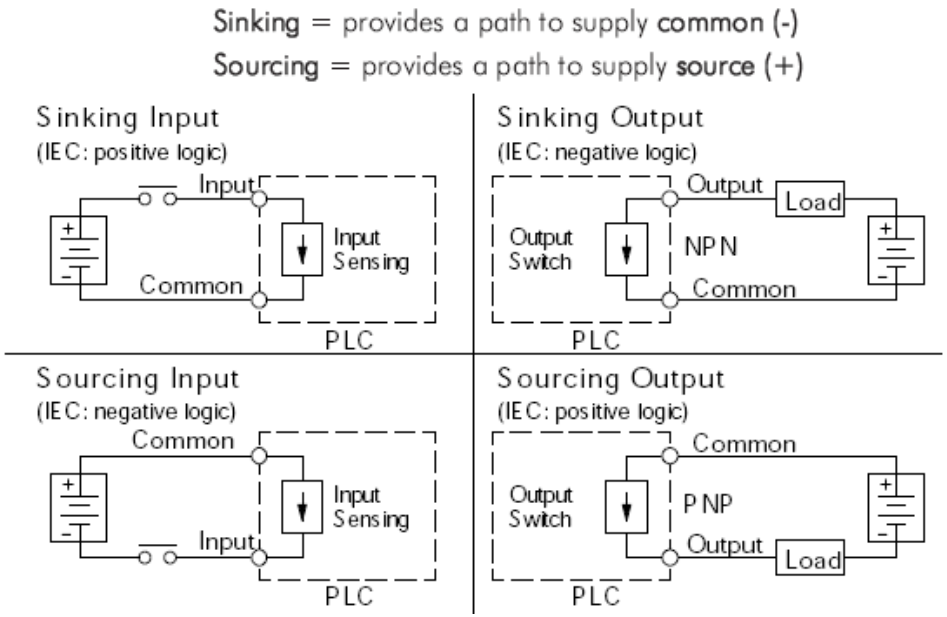
Puis je brancher une ampoule 50W sur une patte de µC ?

Limites de tensions et de courants pour les I/Os

- Courant injecté par un circuit externe dans une des pattes du µC (sinking current), courant fourni par le µC pour « alimenter » un composant externe (sourcing current).



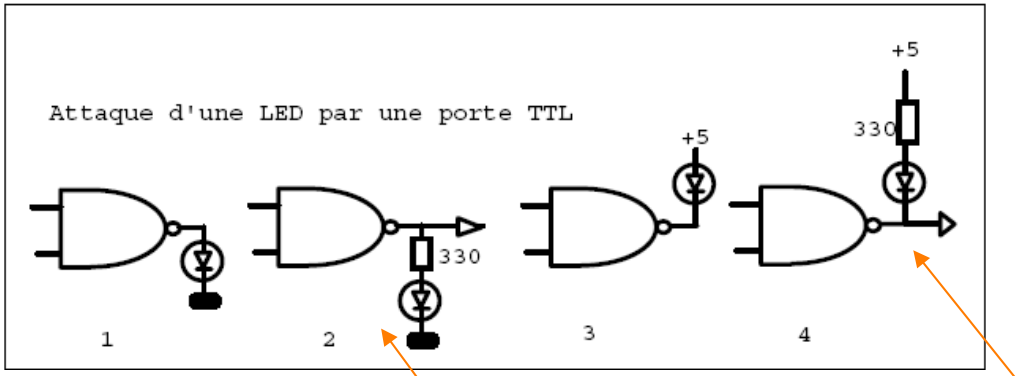




CD5:PIC/sinksource.pdf

Limites de tensions et de courants pour les I/Os

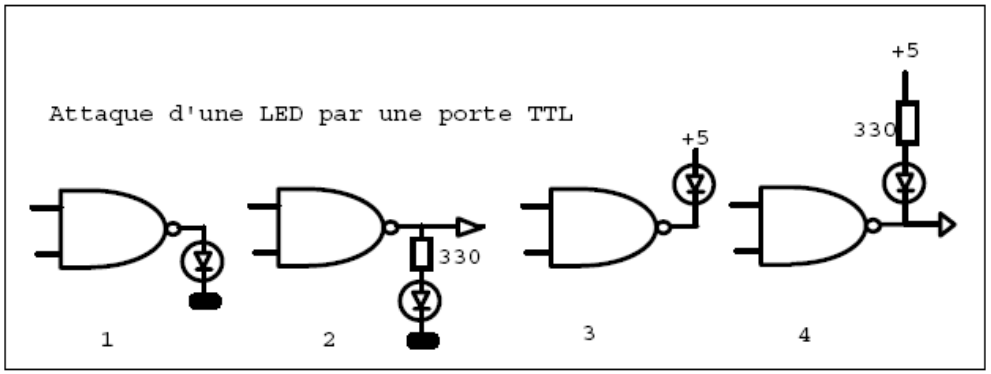
- Limites en courants et en tensions des I/Os ?
- Nous avons vu qu'une I/O peut absorber du courant, ou en fournir.



- Pour prévoir s'ils sont correctement câblés : il faut étudier leurs caractéristiques de sortie Vs=f(Is) ⇨ datasheet
- Vérification à 2 niveaux :
  - De la « patte » I/O du µC : peut elle fournir assez de courant ??? Ou supporter le courant qui va circuler dans la charge (effet Joule R.I²) ?

Max. current out of V_{SS} pin	80 mA
Max. current into V_{DD} pin	80 mA
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{DD}$)	±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{DD}$)	±20 mA
Max. output current sunk by any I/O pin	25 mA
Max. output current sourced by any I/O pin	25 mA
Max. output current sourced by I/O port	75 mA
Max. output current sunk by I/O port	75 mA

Limites de tensions et de courants pour les I/Os



- 2eme niveau :
  - la « charge » a-t-elle assez de courant pour fonctionner ? Ou a-t-on trop de courant (risque de destruction ???)

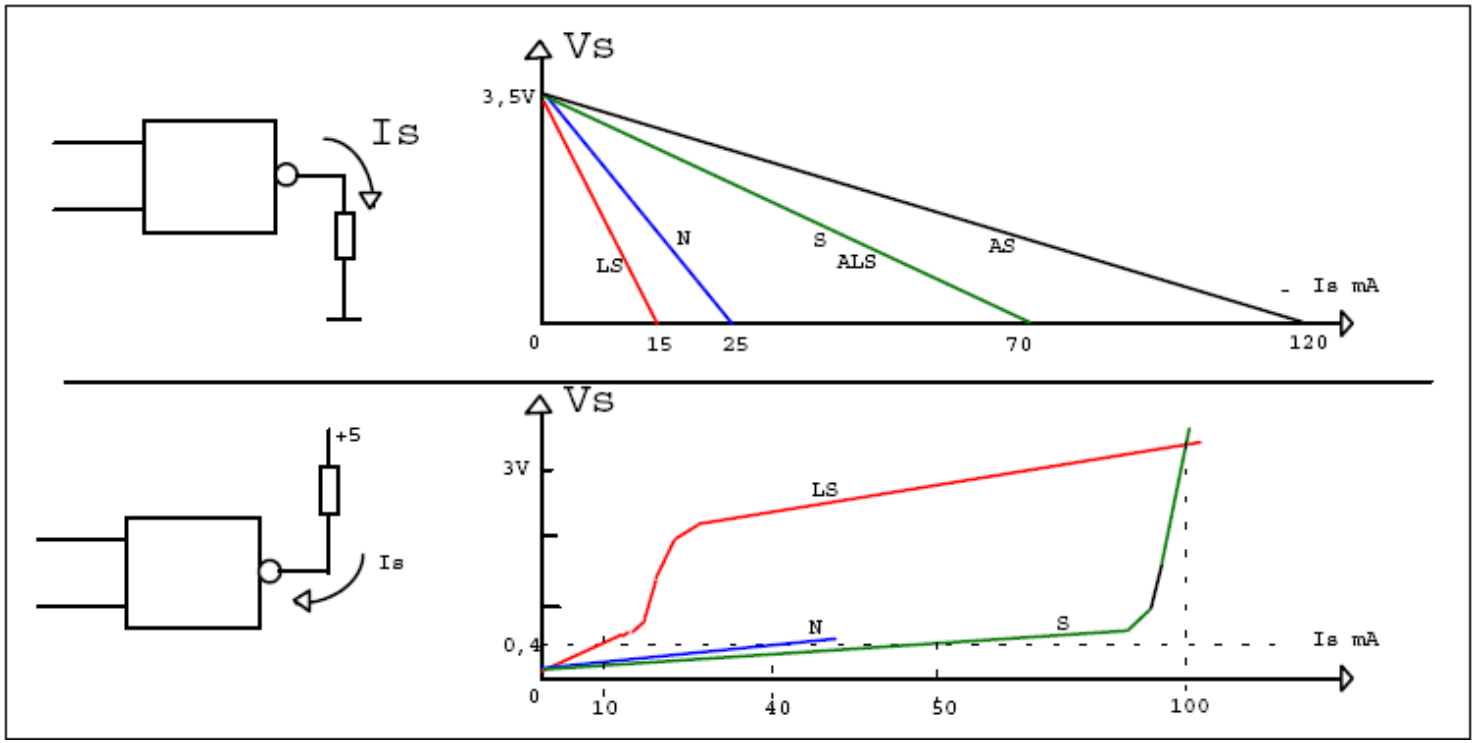
Forward Voltage [4]	V_F	-	3.2	3.8	V

DC Forward Current	I_F	30	mA
Forward Peak Pulse Current	I_{FP} [5]	100	mA

[5] $t \leq 0.1$ms, $D = 1/10$

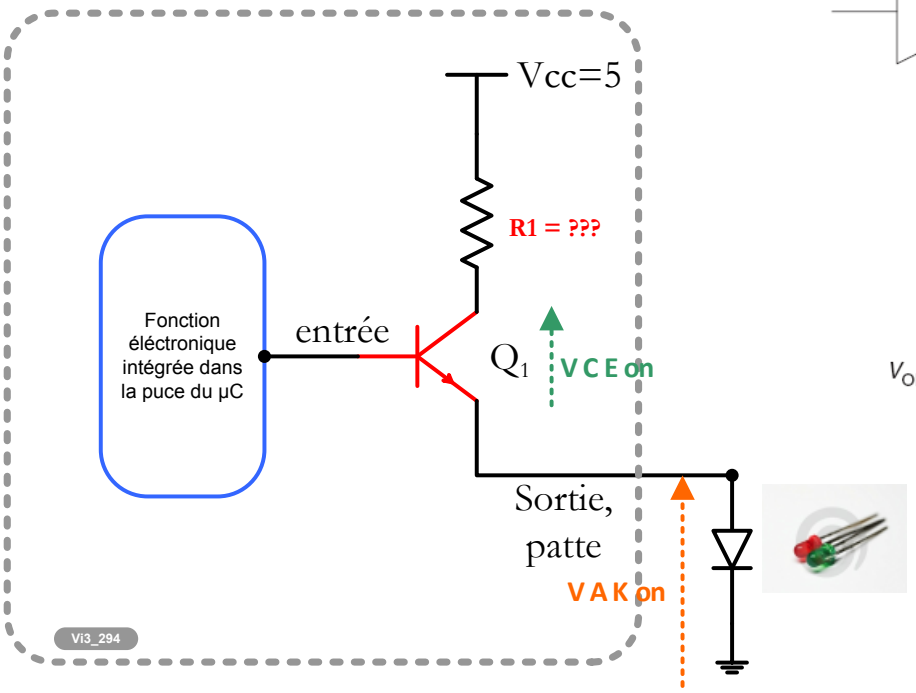
Limites de tensions et de courants pour les I/Os

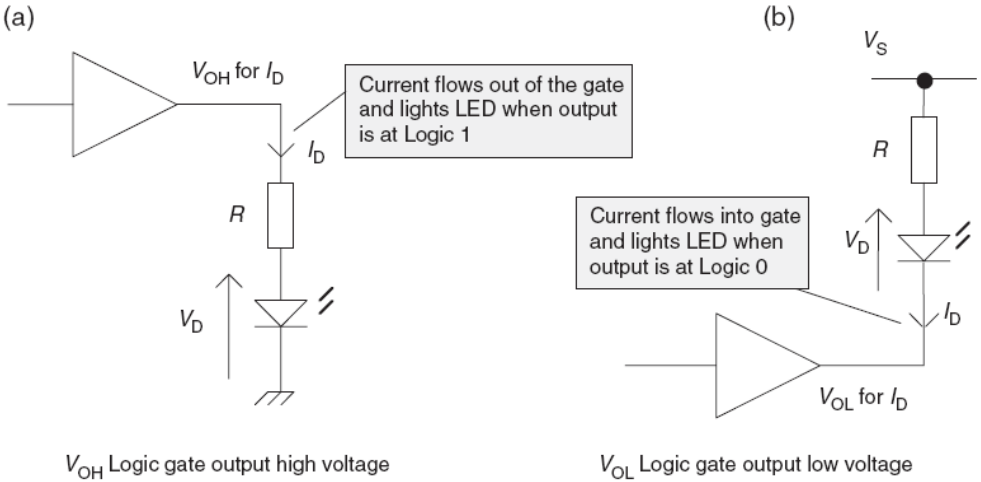
- Datasheet :
  - courbes de caractéristiques de sortie du circuit considéré.
  - courant de court circuit d'une porte LS = 15mA , 25 mA pour une TTL N .
  - Une LED rouge placée à la sortie d'une porte LS est parcourue par une dizaine de milliampéres, elle est par contre détruite si la porte est une S ou AS.



Limites de tensions et de courants pour les I/Os

- Un petit calcul est souvent nécessaire pour $R_{PROTECTION}$ ou I_{FOURNI} :
  - Connaissant Vcc, VCEon, VAKon et la valeur des courants limites (lue dans la datasheet),
  - quelle valeur de R1 choisir ??? ⇨ Loi des Mailles, Loi des Nœuds.





For current source: $V_{OH} = RI_D + V_D$

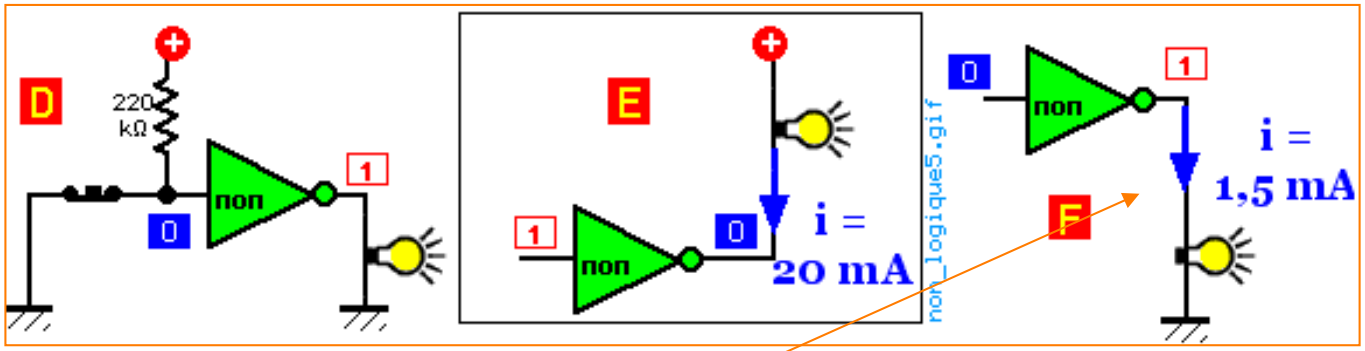
$$R = \frac{V_{OH} - V_D}{I_D}$$

For current sink: $V_S = V_{OL} + RI_D + V_D$

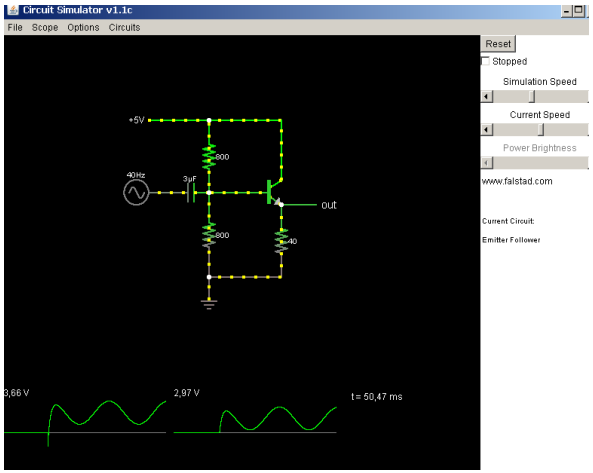
$$R = \frac{V_S - V_D - V_{OL}}{I_D}$$

Limites de tensions et de courants pour les I/Os

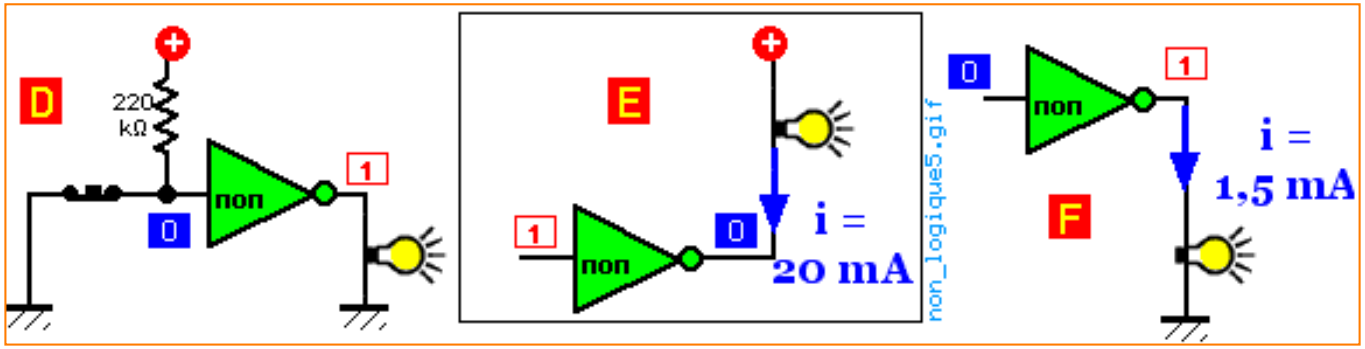
- Question : CD5/interrupteur elec:
- Est-ce logique de trouver dans ces configurations 20mA et 1.5mA ?
- Pourquoi ? D'où vient l'énergie ?



- Si après calcul rapide (KVL, KCL), je me rends compte que je n'ai pas assez de courant en sortie, quoi tenter (⇩)?



- Si après calcul rapide (KVL, KCL), je me rends compte que je n'ai pas assez de courant en sortie, quoi tenter (⇩)?



- Utilisation de plusieurs sorties ensemble (rem : voir configuration sortie open drain, collector, push pull…cf suite cours) :