

Le microprocesseur et ses périphériques

Microprocesseur

Un microprocesseur est un séquenceur qui exécute en permanence les instructions d'un programme codé en assembleur, basé sur un organigramme. Les langages plus évolués, basés sur un arbre programmatique, sont traduits en code machine par un compilateur.

Codage des nombres

Avec n bits, on peut coder 2^n valeurs :

- Avec 10 bits, on peut coder 1024 (Kilo) valeurs.
- Avec 20 bits, on peut coder 1.048.576 (Méga) valeurs.
- Avec 30 bits, on peut coder 1.073.741.824 (Giga) valeurs.

Avec 8 bits, on peut coder 256 valeurs

Avec 16 bits, on peut coder $2^6 \times 2^{10} = 64 \text{ Kilo} = 65536$ valeurs

Avec 32 bits, on peut coder 2^2 Giga valeurs...

Les processeurs permettent le codage de réels, grâce à une mantisse M et un exposant : $N = M \times 2^n$.

La valeur absolue de M est comprise entre 1 et 2 exclus.

Le bit de poids $2^0 = 1$ est sous-entendu, et remplacé par un bit de signe.

Nombres signés

Les entiers positifs ou négatifs, sont représentés en code complément à deux :

Pour passer d'un nombre à son opposé, il suffit de complémenter tous les bits, puis d'ajouter 1.

Démonstration : $dcba + d/c/b/a + 1 = 2^n \equiv 0$!

Ce codage permet de faire aussi bien des additions, que des soustractions avec le même circuit.

Si le premier bit est 0, le nombre est positif ou nul. Si le premier bit est 1, le nombre est strictement négatif.

Voici les entiers représentés avec 4 bits :

-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Masquages :

On utilise un OU pour forcer certains bits à 1 :

1001.0100

+1100.0000

=1101.0100

De même, on utilise un ET pour forcer certains bits à 0.

Décalages :

On décale les bits à gauche pour multiplier par 2 :

$$5 = 0101 \Rightarrow 1010 = 10$$

Conversion binaire / hexadécimal:

Les chiffres binaires sont regroupés en paquets de quatre bits, de la droite vers la gauche.

Un paquet dcba vaut $8d+4c+2b+1a$, soit une valeur comprise entre 0 et 15.

Les valeurs 10 à 15 s'écrivent avec les lettres A à F. Ex : $0111.1010 = 7A$

Conversion hexadécimal/décimal :

A chaque fois qu'on se déplace vers la gauche, on multiplie le code héra par 16.

$$\text{Ex : } 3E8 \text{ h} = 3 \times 256 + 14 \times 16 + 8 = 1000 \text{ d}$$

Conversion décimal/hexadécimal :

De 1000 d, on soustrait $3 \times 256 = 768$, il reste 232.

puis on divise par 16 : $232 = 14 \times 16 + 8$

Autre méthode :

$$1000 = 62 \times 16 + 8 \text{ avec } 62 = 3 \times 16 + 14$$

$$1000 = [3 \times 16 + 14] \times 16 + 8 = 3 \times 256 + 14 \times 16 + 8.$$

Code BCD :

C'est un code binaire qui représente les entiers de 0 à 9

pour additionner deux codes BCD, il faut corriger le résultat :

$$\begin{aligned} &45 \\ &+ 56 \\ &= 9B \\ &+ 06 \\ &= A1 \\ &+ 60 \\ &= 101 \end{aligned}$$

Mémoires :

Mémoires mortes :

Ce sont des mémoires permanentes.

ROM : Read Only Memory. Le contenu est programmé pour toujours à l'usine.

PROM : Programmable une seule fois par l'utilisateur (à jonctions ou à fusibles).

EPROM : Effaçable aux ultraviolets.

EEROM : Electroniquement effaçable, mais la programmation est longue.

Flash : EEROM plus rapide, mais qui dure moins longtemps.

Mémoires vives :

Elles ont besoin d'une alimentation pour garder les données.

NOVRAM : Mémoire non volatile (avec batterie intégrée).

SRAM : Mémoire statique (bascules), très rapide mais la capacité de stockage est moyenne.

DRAM : Mémoire dynamique (capacités), bonne capacité mais à rafraîchir en permanence.

Mémoire cache :

La mémoire cache des microprocesseurs est une mémoire interne plus petite, mais plus rapide que la mémoire externe. Le principe d'une cache programme est le suivant :

Si l'instruction demandée est dans la cache, elle est prise dans la cache.

Sinon, elle est prise dans la mémoire externe, et recopiée dans la cache.

Quand le cache est rempli, et qu'un nouveau bloc arrive, le plus simple est de supprimer le plus anciennement chargé, mais il est plus efficace de supprimer le plus anciennement utilisé.

(Localité spatiale et temporelle).

Si une boucle est plus petite que la taille de la cache, son exécution sera accélérée.

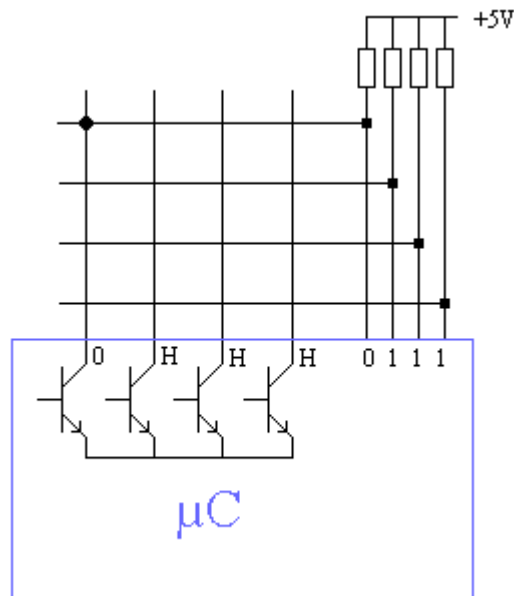
Par contre, si la boucle est plus grande, en gérant la cache de cette manière, elle sera inutile.

Décodage d'adresse :

Si une RAM est sélectionnée par $A15 = 0$ et $A14 = 1$ et $A13 = 1$:

Elle s'étend de 6000 à 7FFF car $(0110) = 6$ et $(0111) = 7$, sur une zone de 8Ko (13 bits).

Lecture d'un clavier :



Pour éviter un court-circuit quand on appuie sur 2 touches, on utilise des sorties à collecteur ou à drain ouvert. Si le μC n'en a pas, on place 0000.0000 dans le tampon de sortie :

- en entrée = Haute impédance (H)
- en sortie = 0

Pour avoir un état haut quand aucune touche n'est appuyée, ou en cas de sortie haute impédance, on met des résistances de rappel.

Il suffit de mettre à 0 une sortie pour lire une ligne ou une colonne, et de lire un 0, pour savoir quelle touche à été enfoncée.

Systèmes à microprocesseur

Le microcontrôleur contient un microprocesseur, de la mémoire et autres périphériques.

L'automate programmable industriel (API), se programme avec un langage particulier, basé sur un grafcet, qui permet de traduire des attentes d'événements...

Pour éviter de faire planter le système, avec des micro-coupures d'alimentation, il est souhaitable, soit de mettre une batterie, soit de détecter une baisse de tension en amont du microprocesseur...

Afficheur 7 segments

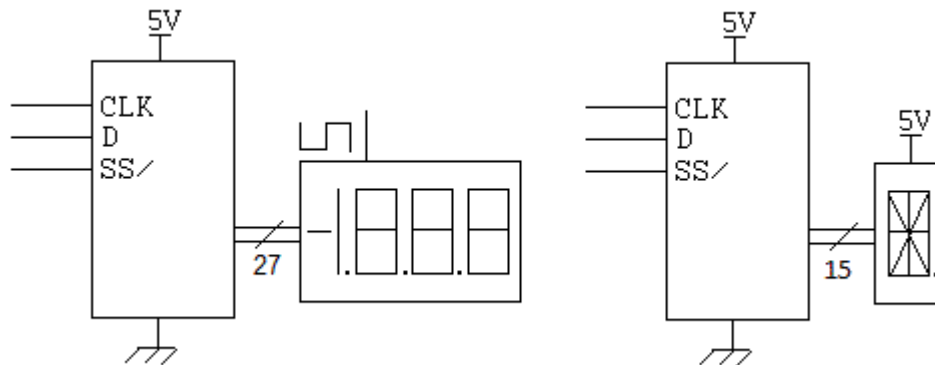
Les afficheurs à leds peuvent être multiplexés par 2 à 4 (à chaque instant, un seul est alimenté).
La persistance rétinienne donne l'illusion que tous les afficheurs sont figés.
Ne pas dépasser 100mA crête par segment.

Avec 2 anodes communes, 2 npn BC337 avec leur collecteur à +6V, le μ C alimenté par un LF50.
Avec 4 cathodes communes, 2 CD4511 alimentés en 6V, les afficheurs multiplexés avec 2 ZXMN3.
Si μ C et afficheurs alimentés en 5V, déclencher la conversion au milieu d'un intervalle (5V parasité)
Les 4511 ont un latch et peuvent être mis après des compteurs décimaux (ex : fréquencemètre).

Codes pour 0 à F : FC, 60, DA, F2, 66 (26), B6, BE, E0, FE, F6, EE, 3E, 9C, 7A, DE, 8E + LSB (.)
GhiJ= BC, 6E, 08, 70. Lmno= 1C, EC, 2A, 3A. Pqrt= CE, E6, 0A, 0E. uWy ρ] = 38, 7C, 76, C6, F0.

CD4543, AY0438, ICM7211, ICM7224, ICL7106 = cristaux liquides. ICM7217 = LED.
STP08DP05, STP16CP05, MM5450, ICM7212, ICL7107 = anode commune.

2 idées futuristes pour LCD et LED, avec temps mort pendant la transmission :



Interface analogique

Pour filtrer une PWM, on peut utiliser 2 filtres RC en cascade avec une fréquence de coupure au moins 30x plus faible.

Les 2 filtres doivent avoir la même fréquence de coupure (-3dB) : $RCw = 1$ donc $f_c = 1/(2\pi RC)$.
Pour que le second filtre n'influe pas le premier, la résistance du second doit être de l'ordre de 5 à 10 fois supérieure (avec le condensateur d'autant plus faible). L'avantage du double filtre est la rapidité pour une même ondulation. Prévoir éventuellement une zener 5.6V (effet d'avalanche) par entrée.

Des amplificateurs opérationnels peuvent réaliser plusieurs fonctions. Les plus récents peuvent recevoir et fournir une tension proche des points d'alimentation (rail to rail, ex TS922), sinon se référer aux 'common mode' et 'output swing'.

Interface de puissance

Le μ P peut piloter un TPIC6C595, TPL7407L, un ULN2003 ou ULN2803 (+économique).
En pont le L293 ou L6207. En High side avec 2V de perte, le MIC2981 ou MIC5891 (SPI).

Pour une lampe à incandescence 230V, un triac BT131, Z0103 (1A), Z0405, TIC206 (sensible 4A) :

Avec une alimentation capacitive négative ou symétrique (33 nF par mA).

Zero-cross : $Z+ = \text{Sign.Hyst}$ / $Z- = \text{Sign.Hyst}$ $I+ = D-.Z+$ / $I- = D+.Z-$

Ou moins dangereux avec transfo et optotriac MOC30 ou relais statique AQH32.

Pour piloter un moteur pas à pas, le mode demi pas permet d'éviter les vibrations ou le décrochage.

Allumage crépusculaire anti intempestif : $R = \text{On}/.Qt$ $S = \text{On}.Qt/$