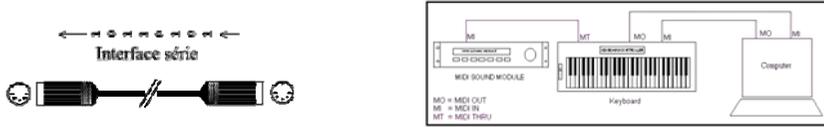
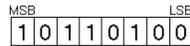


Il est important de savoir qu'avec le protocole MIDI, aucun signal audio n'est transmis. Avec ce protocole, on ne *transmet que des commandes codées en base hexadécimale* (exemples : déclenchement et l'arrêt d'une note de musique, le choix d'un son de type piano ou flute, l'utilisation de la pédale de soutien, le "pitch bend"). Ces codes sont transmis via une interface série *sous forme binaire* sur un câble MIDI comportant des connecteurs DIN 5 broches (cf ci-dessous)



Format de message : la plupart des messages MIDI se composent de **un, deux ou trois octets** (mots). Chaque octet peut représenter une commande ou une donnée.

Un décodeur MIDI examine toujours le bit le plus significatif de la trame envoyée pour voir sa valeur 1 ou 0. Les **octets de commandes**, appelés aussi « mots d'état », commence tous par un 1, alors que les **octets de données** commence par un 0.



L'octet de commande est le premier mot d'un message MIDI. Il sert d'instruction qui, une fois interprétée par le décodeur, déclenchera un **événement MIDI**. Les octets suivants sont des données qui viennent fournir les paramètres de la commande

On souhaite transmettre 1 octets de commande et 2 octets de données pour réaliser l'évènement midi « **déclenchement d'une note MIDI** » décrit dans le tableau ci-dessous.

Remarque : lors de leur transmission, les codes MIDI sont d'une longueur de 10 bits : **8 bits (donc 1 octet) d'information MIDI plus un bit de départ et 1 bit d'arrêt**. Les bits de départ et d'arrêt valent toujours zéro et ne servent qu'à synchroniser le transfert des données entre les différents modules du réseau MIDI. Ces deux bits font partie des spécifications de transfert des données MIDI et ne contiennent aucune information musicale.

| | |
|-------------------|--|
| 9 0 ₁₆ | Octet de commande – Déclenchement d'une note sur le canal MIDI 1. |
| 3 C ₁₆ | Octet de donnée – Le code de la note à déclencher. C'est la 60 ^{ième} note des 128 possibles (0 à 127) correspondant au « do » du milieu du clavier. |
| 7 2 ₁₆ | Octet de donnée – Le code de vélocité indiquant la vitesse à laquelle la note a été frappée. Le domaine des valeurs possibles varie de 0 à 127. Cette valeur contrôle le volume du son |

1. Donner le code binaire associé à cet évènement de trois mots.

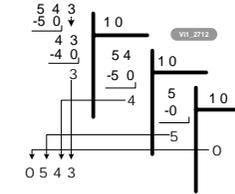
2. Soit le code binaire ci-dessous sur 2 octets traduisant l'évènement « sélection de son sur la canal n°16 ». Donner sa représentation en hexadécimal.



VII. Systèmes de numération

Pour revoir la base 2 et l'algorithme de conversion associé :

- Ecrire 173₁₀ en binaire (en présentant votre conversion sous forme de divisions successives par 2 « méthode des divisions successives » - voir l'exemple proposé en base 10 ci-dessous).
- Puis passer du binaire à l'octal et à l'hexadécimal.
- Rappeler brièvement le principe de passage de la base binaire à octal et à hexadécimal.



VIII. Base binaire, octal, hexadécimal

- Donner la valeur décimale du nombre 10101, dans le cas où il est codé en base 2, 8 et 16.

IX. Codes détecteurs d'erreurs

Ces codes peuvent être obtenus par ajout

- d'un **digit de parité** ou de contrôle de façon à avoir un nombre pair de bits à l'état 1 (la parité est alors dite paire) ou impair dans l'autre cas.
- Ou d'un **caractère de contrôle** après un certain nombre de bits transmis de tel façon à avoir par colonne un nombre de bits à l'état 1 pair (la parité est alors dite paire) ou impair.
- Compléter les tableaux suivants dans le cas d'une parité paire :

| information binaire a transmettre | bit de parité | |
|-----------------------------------|---------------|---|
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 1 | 0 | 1 |

exc1_2631

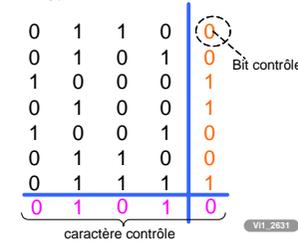
Un bit de parité est transmis après transmission de 3 digits d'information

| information binaire a transmettre | caractère de contrôle | |
|-----------------------------------|-----------------------|---|
| 0 | 1 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 1 |

exc1_2631

Un mot de 4 digits est transmis après la transmission de 5 mots d'information

- Un **code correcteur (à la différence d'un code détecteur)** doit non seulement détecter une erreur mais la localiser ! d'où l'utilisation conjointe des deux techniques précédemment présentées (mise en œuvre d'une détection de ligne-colonne, x-y).



X. Opérations en binaire

1. Réaliser la somme en base 2 des nombres décimaux suivants : 25 et 14
2. Même question avec 31 et 14.
3. réaliser en décimal (en posant l'opération) $2272 - 891$
4. réaliser en binaire (après avoir réalisé la passage en base 2) les soustractions « 4-3 » et « 19-5 »

XI. Addition et soustraction en hexadécimal

1. Additionner en hexadécimal (comme vous l'auriez fait en décimal), $9C4_{16}$ et DAC_{16} .
2. Vérifier votre résultat en base 10.
3. Soustraire en hexadécimal $9F$ et $1C$
4. Soustraire en hexadécimal $E19F$ et $C7E6$

XII. Multiplication binaire et hexadécimale

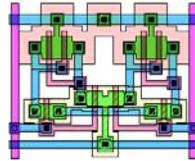
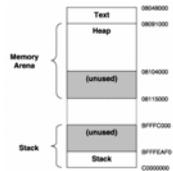
1. Multiplier en décimal $33_{10} * 11_{10}$. Présenter votre calcul en explicitant la notion de décalage associée à la multiplication de nombres de plusieurs chiffres. Réaliser $11_{10} * 6_{10}$ en base 2.
2. En terme d'algorithme, qu'elles sont les deux opérations mises en œuvre pour avoir le résultat de cette multiplication ?
3. réaliser en décimal en observant bien votre méthode de calcul : $3578 * 13$
4. réaliser en hexadécimal : $B4 * 13$
5. Quelle est la principale opération que vous utiliser pour réaliser une multiplication.

XIII. Divisions

1. réaliser en décimal la division $1673/3$
2. réaliser en binaire la division $1101/10$

XIV. Combinaisons

1. En supposant un codage classique binaire sur n bits. Combien de valeurs distinctes peuvent être représentées. Rappeler la plus grande valeur représentable.



2. Combien de digits binaires (=nombre de pattes d'une puce mémoire) faut-il pour pouvoir adresser 65 563 mots mémoire différents ?

XV. Rappel multiplication – perte d'informations

Dans la notation que l'on nomme « représentation à virgule fixe », la position de la virgule est fixe par rapport par rapport aux nombres de digits qui composent le nombre traité (exemple « trois chiffres après la virgule »).

0 0 1 1 0 0 1 1 1 0

Dans un calculateur cela peut poser problème : illustrer et commenter le problème avec un calculateur en base 10 à 1 chiffre après la virgule et la multiplication en base 10 de « $17,41 * 3,10$ ». Le problème se pose-t-il aussi en base 2 ?

XVI. Codage binaire, octal – addition

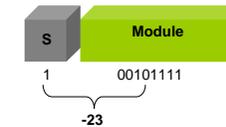
1. Coder en binaire sur un octet les entiers 105 et 21 puis effectuer l'addition binaire des entiers ainsi codés. Vérifier que le résultat sur un octet est correct.
2. Mêmes questions avec les entiers 184 et 72.

XVII. Nombre binaire non-entier

Soit le réel « 0,546875 » en base 10. Traduisez le directement en base 2.

XVIII. Nombre binaire négatif – signe valeur

En considérant un codage binaire de nombre « signe-valeur » à virgule fixe, donnez la valeur en base 10 des nombres binaires 1101 0101, 1001 0001. La notation « signe-valeur » est présentée ci-dessous :



XIX. Nombre binaire négatif – complément à 2

Indiquer la valeur codée par la suite 1101 1001 0111 0101 qui représente un entier signé en complément à 2 sur 16 bits.

XX. Notation binaire virgule flottante

Soient les réels « 512 » « 1.875 » et « 1.9375 » en base 10. Donner leur représentation en binaire à virgule flottante simple précision. Le traduire en hexadécimal.



PHYS 246 A Architecture μ C – FICHE TD2

I. Regroupement des digits – conversion de base 2 et 8

1. Ecrire 113_{10} en binaire (en présentant votre conversion sous forme de divisions successives par 2 « méthode des divisions successives »).
2. Même exercice avec la base octal.
3. Retrouver les équivalences entre les 2 représentations du même nombre décimal.
4. Enoncer la méthode pour passer de la base 2 à la base 8.
5. Enoncer la méthode pour passer de la base 8 à la base 2.

II. Numération et conventions

Parmi la liste de nombres suivante : {1001, 52, AE01, 12487, 43, 110, 2}, donner :

1. Groupe 1 : les nombres qui peuvent être écrits en base 5 OU décimal.
2. Groupe 2 : les nombres qui sont en hexadécimal OU binaire.
3. Groupe 3 : les nombres qui peuvent être en base 5 ET en décimal.

III. Possibilité de codage complément à 2 (C2)

Soit « une entrée » de calculateur sur 4 digits. Ce calculateur utilise la convention de notation complément à 2.



1. Donner l'intégralité des codes possibles en entrée du calculateur.
2. Donner leur équivalent décimal
3. A partir du nombre de digits en entrée, trouver le nombre de nombres différents représentés, le plus grand (en valeur absolue) nombre négatif représentable en C2, le plus grand nombre positif représentable en C2.
4. Généraliser à « n digits » en entrée.

IV. Codage BCD

La notation BCD (Binary Coded Decimal) est utilisée dans certains circuits.

6. Donner en BCD la représentation du nombre décimal 128.
7. Le code binaire suivant est-il codé en BCD 0010 1101 0100 ?
8. Réaliser en convention BCD $(40+35)_{10}$? Le résultat est-il cohérent ?
9. Réaliser en convention BCD $(18+16)_{10}$? Le résultat est-il cohérent ?
10. Pour retrouver un résultat correct de l'addition en BCD, vérifier alors la méthodologie suivante « effectuer d'abord l'addition en motifs de 4 bits. Puis si le résultat dépasse 9 pour l'un ou plusieurs de ces motifs, on leur ajoute 6 pour forcer une retenue ».
11. Réaliser en BCD la somme : $(76+45)_{10}$

V. Complément à 2

Démontrer que le complément à 2 représente bien l'opposé d'un nombre à partir des éléments vu en cours sur cette notation ?

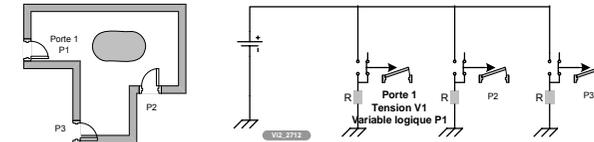
| (N)10 | (N)2 | N | (-N)2 | (-N)10 |
|-------|------|------|-------|--------|
| 7 | 0111 | 1000 | 1001 | -7 |
| 3 | 0011 | 1100 | 1101 | -3 |

VI. Nombre binaire à virgule – équivalence et précision

Soit le nombre 477.527_{10} . Quelle précision (\Leftrightarrow nombre chiffres après la virgule) doit-on retenir pour avoir une représentation correcte en base 2 ?

VII. Circuit surveillance – table vérité

Soit un bâtiment équipé de 3 portes : P1, P2, P3.



Nous avons installé dans un bâtiment (voir le schéma de la pièce), un circuit qui détecte quand 2 portes sont fermées simultanément. En effet il doit toujours y avoir une circulation d'air dans cette pièce. Quand 2 portes sont fermées alors la sortie S doit passer à 1 et allume par la même une LED de visualisation.

1. A quelle valeur binaire correspond le fait d'avoir « la porte fermée » ?
2. Réaliser et compléter le tableau admettant comme colonne P1 P2 P3 et S

Remarque : la table que vous venez de construire s'appelle table de vérité

VIII. Représentation en binaire flottant simple précision

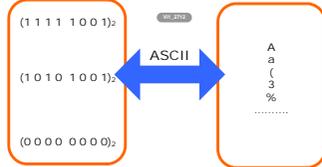
Soit les nombres décimaux suivants : 3.4 et 20.8×10^1 :

1. Donner leur représentation en binaire flottant simple précision.
2. Coder leur somme en simple précision.
3. Peut-on sommer directement les mantisses et exposants ?

IX. Code ASCII

L'information est souvent stockée sous forme numérique dans les ordinateurs. Il n'existe pas de méthode pour stocker directement les caractères, lettres, marques de ponctuation tels que &, @, (, \.

Il a donc été créé un codage particulier pour ces symboles compréhensibles par l'humain. Chaque caractère possède son équivalent en code numérique : c'est le code ASCII (American Standard Code for Information Interchange).



Le code **ASCII de base ou standard** représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127).

1. Donner en hexadécimal le code du 1^{er} symbole et du dernier symbole du code ASCII standard.

| DEC | HEX | CHAR |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 0 | 00 | NUL | 32 | 20 | SP | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 01 | SOH | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 02 | STX | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 03 | ETX | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 04 | EOT | 36 | 24 | \$ | 68 | 44 | D | 100 | 64 | d |
| 5 | 05 | ENO | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 06 | ACK | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 07 | BEL | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 08 | BS | 40 | 28 | (| 72 | 48 | H | 104 | 68 | h |
| 9 | 09 | HT | 41 | 29 |) | 73 | 49 | I | 105 | 69 | i |
| 10 | 0A | LF | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | 0B | VT | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | 0C | FF | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | 0D | CR | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | 0E | SO | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | 0F | SI | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | DLE | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | DC1 | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | DC2 | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | DC3 | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | DC4 | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | NAK | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | SYN | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | ETB | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | CAN | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | EM | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | SUB | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | ESC | 59 | 3B | ; | 91 | 5B | [| 123 | 7B | { |
| 28 | 1C | FS | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | |
| 29 | 1D | GS | 61 | 3D | = | 93 | 5D |] | 125 | 7D | } |
| 30 | 1E | RS | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | US | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | DEL |

Les codes **0 à 31** ne sont pas des caractères. On les appelle **caractères de contrôle** car ils permettent de faire des actions (code de contrôle associée à la communication, caractères non imprimables tabulation, retour à la ligne...) telles que :

- retour à la ligne (CR)
- SYN=22₁₀=16₁₆= synchronous idle
- ETB=23₁₀=17₁₆= end of trans. block

Les codes **65 à 90** représentent les **majuscules**
 Les codes **97 à 122** représentent les **minuscules**

2. Donner le code hexadécimal correspondant à la lettre « M » et celui de la lettre « m ». Donner alors leur correspondance en binaire.
3. Même question pour « Z » et « z ».
4. Par rapport aux 2 questions précédentes que remarquez vous du passage majuscule ↔ minuscule ?

Le code ASCII a été mis au point pour la **langue anglaise**, il ne contient donc pas de **caractères accentués**, ni de caractères spécifiques à une langue. Pour coder ce type de caractère il faut recourir à **un autre code**. Le code ASCII a donc été **étendu à 8 bits (un octet)** pour pouvoir coder plus de caractères (on parle d'ailleurs de **code ASCII étendu**).

| DEC | HEX | CHAR |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 128 | 80 | À | 160 | A0 | NBSP | 192 | C0 | À | 224 | E0 | à |
| 129 | 81 | Á | 161 | A1 | í | 193 | C1 | Á | 225 | E1 | á |
| 130 | 82 | Â | 162 | A2 | î | 194 | C2 | Â | 226 | E2 | â |
| 131 | 83 | Ã | 163 | A3 | ë | 195 | C3 | Ã | 227 | E3 | ã |
| 132 | 84 | Ä | 164 | A4 | ü | 196 | C4 | Ä | 228 | E4 | ä |
| 133 | 85 | Å | 165 | A5 | ¥ | 197 | C5 | Å | 229 | E5 | å |
| 134 | 86 | Æ | 166 | A6 | ¦ | 198 | C6 | Æ | 230 | E6 | æ |
| 135 | 87 | Ç | 167 | A7 | § | 199 | C7 | Ç | 231 | E7 | ç |
| 136 | 88 | È | 168 | A8 | ¨ | 200 | C8 | È | 232 | E8 | è |
| 137 | 89 | É | 169 | A9 | © | 201 | C9 | É | 233 | E9 | é |
| 138 | 8A | Ê | 170 | AA | ª | 202 | CA | Ê | 234 | EA | ê |
| 139 | 8B | Ë | 171 | AB | « | 203 | CB | Ë | 235 | EB | ë |
| 140 | 8C | Ì | 172 | AC | » | 204 | CC | Ì | 236 | EC | ì |
| 141 | 8D | Í | 173 | AD | - | 205 | CD | Í | 237 | ED | í |
| 142 | 8E | Î | 174 | AE | ® | 206 | CE | Î | 238 | EE | î |
| 143 | 8F | Ï | 175 | AF | ™ | 207 | CF | Ï | 239 | EF | ï |
| 144 | 90 | Ð | 176 | B0 | ° | 208 | D0 | Ð | 240 | F0 | ð |
| 145 | 91 | Ñ | 177 | B1 | ± | 209 | D1 | Ñ | 241 | F1 | ñ |
| 146 | 92 | Ò | 178 | B2 | ² | 210 | D2 | Ò | 242 | F2 | ò |
| 147 | 93 | Ó | 179 | B3 | ³ | 211 | D3 | Ó | 243 | F3 | ó |
| 148 | 94 | Ô | 180 | B4 | ´ | 212 | D4 | Ô | 244 | F4 | ô |
| 149 | 95 | Õ | 181 | B5 | µ | 213 | D5 | Õ | 245 | F5 | õ |
| 150 | 96 | Ö | 182 | B6 | ¶ | 214 | D6 | Ö | 246 | F6 | ö |
| 151 | 97 | × | 183 | B7 | · | 215 | D7 | × | 247 | F7 | × |
| 152 | 98 | Ø | 184 | B8 | ¸ | 216 | D8 | Ø | 248 | F8 | ø |
| 153 | 99 | Ù | 185 | B9 | ¹ | 217 | D9 | Ù | 249 | F9 | ù |
| 154 | 9A | Ú | 186 | BA | º | 218 | DA | Ú | 250 | FA | ú |
| 155 | 9B | Û | 187 | BB | » | 219 | DB | Û | 251 | FB | û |
| 156 | 9C | Ü | 188 | BC | ¼ | 220 | DC | Ü | 252 | FC | ü |
| 157 | 9D | Ý | 189 | BD | ½ | 221 | DD | Ý | 253 | FD | ý |
| 158 | 9E | Þ | 190 | BE | ¾ | 222 | DE | Þ | 254 | FE | þ |
| 159 | 9F | ß | 191 | BF | ¿ | 223 | DF | ß | 255 | FF | ÿ |

Ce code attribue les valeurs 0 à 255 (donc codées sur **8 bits, soit 1 octet**) aux lettres majuscules et minuscules, aux chiffres, aux marques de ponctuation et aux autres symboles (caractères accentués dans le cas du code iso-latin1). Les codes ASCII du jeu étendu qui vont de 128 à 255 correspondent à des symboles variés qui dépend du constructeur (attention donc à leur interprétation).

Le code ASCII étendu n'est pas unique et dépend fortement de la plateforme utilisée.

Les deux jeux de caractères ASCII étendus les plus couramment utilisés sont :
 Le code ASCII étendu OEM, c'est-à-dire celui qui équipait les premières machines de type IBM PC. Le code ASCII étendu ANSI, utilisé par les systèmes d'exploitation récents.

Sur votre ordinateur, tapez « Alt+ **036** » (sans lâcher a touche ALT) pour afficher un $\$$.

X. Codage binaire signé – « signe-valeur absolue »

Coder les entiers décimaux 61 et -61 sur un octet en utilisant la représentation « signe - valeur absolue ». Montrer que l'addition binaire de ces entiers ainsi codés produit un résultat incorrect. Montrer qu'en revanche le résultat est correct si ces entiers sont codés en utilisant la représentation par le complément à deux.

XI. Représentation binaire – complément à 2 – règles addition

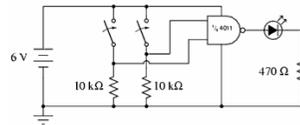
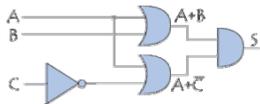
Soit une représentation de nombres négatifs en complément à 2 sur 8 bits. Effectuer en binaire dans cette convention les opérations 1-2, 51+127, -3-127, -127+127, -63-63. Préciser, pour chaque opération, la retenue et si il y a une erreur (erreur d'interprétation du résultat). Enoncer les règles sur le débordement en fonction du signe des 2 opérands à partir des différentes opérations réalisées.

XII. Représentation virgule flottante – somme et multiplication

Soit le nombre a = 0.10010*10¹⁰¹ (le « 101 » est en binaire aussi) en binaire et b = 0.11010*10¹ en binaire aussi. Calculer a+b et a*b



PHYS 246 A Architecture μC - FICHE TD3



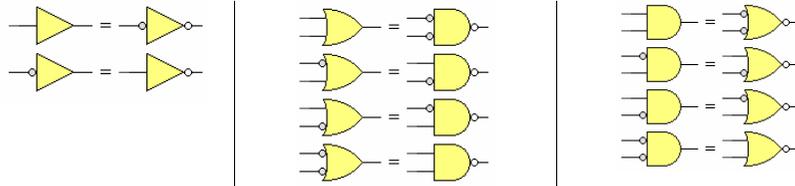
Remarque : il vous est demandé de bien noter et de comprendre les termes techniques utilisés dans les énoncés. Cette démarche fait partie de l'apprentissage scientifique du module.

I. Nombre de codes représentés en complément à 2

En considérant la valeur $n=4$, « vérifiez » qu'en complément à 2, il est possible de représenter les nombres compris entre : -2^{n-1} et $2^{n-1}-1$.

II. Equivalence

Démontrer les égalités suivantes (introduisez des variables logiques de votre choix pour réaliser la démonstration) :



III. Regroupements et compléments

Supprimer la barre d'inversion globale (en respectant bien sûr les règles de l'algèbre de Boole) et développer les expressions au maximum (utiliser judicieusement les parenthèses)

- $F_1 = \overline{\overline{D}y.A}$
- $F_2 = x.z.A\overline{B}$
- $F_3 = A\overline{y}.Z$
- $F_4 = \overline{(x+y)}.z.b$

IV. Groupe logique complet

Soit une fonction NOR à 2 entrées : $\text{NOR}(x,y) = \overline{(x+y)}$. Montrer que la fonction NOR à deux entrées, forme un groupe logique complet. C'est-à-dire que l'on peut réaliser à partir de celui-ci les fonctions de base « non, et, ou » ?

Ex : comment obtenir \overline{A} à partir de $\overline{(x+y)}$.

Quelles sont les conséquences ou avantages associés sur la réalisation pratique des circuits ?

Que dire des groupes suivants : $\{nand\}$, $\{ \cdot, \cdot \} = \{\text{inversion, ET}\}$ et $\{ \cdot, + \} = \{\text{inversion, OU}\}$?

V. Fonction logique de base XOR

On note respectivement les opérateurs **ou**, **et**, **xor** et **non** par $+$, \cdot , \oplus et $\overline{\quad}$. Montrer à l'aide de tables de vérité que $A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$ et que $A \oplus B = (A+B) \cdot (\overline{A} + \overline{B})$

VI. Simplifiez algébriquement les fonctions suivantes :

$$\begin{aligned}
 F_1 &= (x \cdot \overline{y} + z) \cdot (x + \overline{y}) \cdot z \\
 F_2 &= (x + y) \cdot z + \overline{x} \cdot (\overline{y} + z) + \overline{y} \\
 F_3 &= A + A \cdot B \cdot C + \overline{A} \cdot B \cdot C + \overline{A} \cdot B + A \cdot D + A \cdot \overline{D} \\
 F_4 &= (a + b) \cdot (a + b + c) + \overline{a} \cdot \overline{b} + \overline{a} \cdot \overline{c} \\
 F_5 &= (\overline{a} \cdot b + a \cdot \overline{b}) \cdot (a \cdot b + \overline{a} \cdot \overline{b}) \\
 F_6 &= a \cdot b \cdot c + c \cdot (a \cdot \overline{b} + \overline{a} \cdot b) \\
 F_7 &= a \cdot b \cdot \overline{c} + b \cdot (a + \overline{c}) + \overline{a} + b + \overline{a} \cdot c
 \end{aligned}$$

VII. Complémenter puis simplifier les fonctions suivantes :

$$\begin{aligned}
 F_1 &= (a \cdot b) + (b \cdot c) + (a \cdot c) \\
 F_2 &= \overline{a} \cdot \overline{b} + a \cdot b + a \cdot \overline{b} \\
 F_3 &= \overline{c} \cdot \overline{d} + \overline{a} \cdot \overline{b} + c \cdot \overline{d}
 \end{aligned}$$

VIII. Formes canoniques (formes normales)

1/ Deux fonctions sont équivalentes si elles ont la même forme canonique. Démontrer les équivalences suivantes :

$$\begin{aligned}
 T_1 &= (a + b) \cdot (\overline{a} + c) \equiv a \cdot c + \overline{a} \cdot b \\
 T_2 &= (a + b) \cdot (\overline{a} + c) \cdot (b + c) \equiv (a + b) \cdot (\overline{a} + c) \\
 T_3 &= c \cdot d + a \cdot b \cdot c + b \cdot \overline{d} \equiv c \cdot d + b \cdot \overline{d}
 \end{aligned}$$

2/ Mettre les fonctions suivantes sous la forme d'une somme canonique (somme de produits) :

$$\begin{aligned}
 F_1 &= (a + c) \cdot (\overline{a} + b + c) \\
 F_2 &= \overline{a} \cdot b \cdot d + a \cdot \overline{b} \cdot \overline{c} + a \cdot b \cdot c \\
 F_3 &= \overline{a} \cdot \overline{c} + a \cdot b \cdot \overline{d} \cdot c
 \end{aligned}$$

3/ Mettre les fonctions suivantes sous la forme d'un produit canonique (produit de sommes) :

$$\begin{aligned}
 G_1 &= (a + c) \cdot (\overline{a} + b + c) \\
 G_2 &= (a + b + c) \cdot (\overline{a} \cdot b + \overline{c}) \\
 G_3 &= a + b \cdot c + \overline{b} \cdot \overline{c} \cdot d
 \end{aligned}$$

IX. Tables de vérité

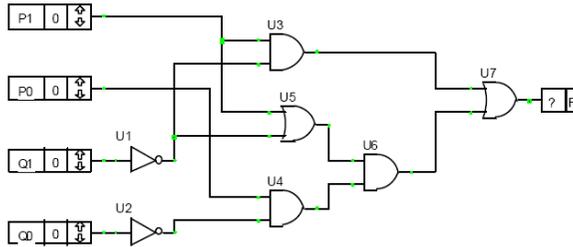
1/ Etablir les tables de vérité des fonctions suivantes :

$$\begin{aligned}
 F_1 &= (\overline{a} \cdot b + a \cdot \overline{b}) \cdot \overline{c} + (\overline{a} \cdot \overline{b} + a \cdot b) \cdot c \\
 F_2 &= a \cdot \overline{b} + b \cdot \overline{c} + c \cdot \overline{a} + a \cdot b \cdot c
 \end{aligned}$$

2/ Etablir la table de vérité et l'expression de la fonction désignée ci-dessous :

On souhaite établir un comparateur de 2 nombres binaires (P, Q) représentés sur 2 bits (P1P0; Q1Q0). La sortie du comparateur vaut 1 si P > Q, 0 sinon.

3/ Etablir la table de vérité de la fonction logique réalisée par le circuit numérique à 4 entrées ci-dessous :



X. Logigrammes

Réaliser les logigrammes des fonctions suivantes avec les circuits numériques proposés :

$$F = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{C} \cdot D$$

avec 3 portes NOR (si besoin, 1 porte NOR \Leftrightarrow 1 porte AND à « 2 entrées inverseuses »)

$$G = A \cdot (B + C)$$

avec 3 portes NAND à 2 entrées

$$H = A \cdot B + B \cdot C + A \cdot C$$

avec des portes NAND à 2 entrées

XI. Expression d'expression algébrique

Démontrer que toute fonction à trois variables F(A,B,C) est égale à $F(A,B,C) = A \cdot F(1,B,C) + \bar{A} \cdot F(0,B,C)$

XII. Tableau de Karnaugh et expression algébrique

Lequel de ces regroupements conduira à l'expression algébrique la plus simple ? Justifier votre réponse.

Par rapport au regroupement que vous avez choisi, existe-t-il un regroupement plus intéressant ? Série A :

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 0 | 0 |
| 11 | | 1 | 0 | 0 | 0 |
| 10 | | 1 | 0 | 0 | 1 |

Table 1

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 0 | 0 |
| 11 | | 1 | 0 | 0 | 0 |
| 10 | | 1 | 0 | 0 | 1 |

Table 2

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 0 | 0 |
| 11 | | 1 | 0 | 0 | 0 |
| 10 | | 1 | 0 | 0 | 1 |

Table 3

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 0 | 0 |
| 11 | | 1 | 0 | 0 | 0 |
| 10 | | 1 | 0 | 0 | 1 |

Table 4

Série B :

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 1 | 1 |
| 11 | | 1 | 0 | 1 | 1 |
| 10 | | 1 | 0 | 0 | 1 |

Table 1

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 1 | 1 |
| 11 | | 1 | 0 | 1 | 1 |
| 10 | | 1 | 0 | 0 | 1 |

Table 2

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 1 | 1 |
| 11 | | 1 | 0 | 1 | 1 |
| 10 | | 1 | 0 | 0 | 1 |

Table 3

| F | YZ | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| 00 | WX | 1 | 0 | 0 | 1 |
| 01 | | 1 | 0 | 1 | 1 |
| 11 | | 1 | 0 | 1 | 1 |
| 10 | | 1 | 0 | 0 | 1 |

Table 4

XIII. Expressions algébriques

Donner l'expression logique associée aux tableaux suivants :

| F | B | C | 00 | 01 | 11 | 10 |
|---|---|---|----|----|----|----|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 4 | 5 | 7 | 6 |

Tableau 1 :

| F | B | C | 00 | 01 | 11 | 10 |
|---|---|---|----|----|----|----|
| 0 | 1 | 0 | 1 | 1 | 3 | 2 |
| 1 | 0 | 0 | 4 | 5 | 7 | 6 |

Tableau 2 :

XIV. Algèbre de Boole

Soit la fonction F définie par la table de vérité suivante :

| A | B | C | F(A,B,C) |
|---|---|---|----------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

- Expression 1 = Exprimez F sous la forme d'une somme de produit. Pour cette question, doit on travailler avec les min-termes ou les max-termes ?
- Expression 2 = Exprimez F sous la forme d'un produit de sommes
- Démontrer par simplification de l'expression 2, que nous retrouvons l'expression 1

XV. Exercice 1

En utilisant les tableaux de Karnaugh, trouver une expression minimale sous la forme d'une somme de produits des fonctions suivantes :

$$F_1 = \sum_{A,B,C} (0,1,2,4)$$

$$F_3 = \prod_{A,B,C} (1,2,6,7)$$

$$F_5 = \sum_{W,X,Y,Z} (1,2,4,7,8,11,13,14)$$

$$F_2 = \sum_{W,X,Y,Z} (1,4,5,6,11,12,13,14,15)$$

$$F_4 = \sum_{W,X,Y,Z} (0,1,2,3,7,8,10,11,15)$$

$$F_6 = \prod_{A,B,C,D} (1,3,4,5,6,7,9,12,13)$$

XVI. Exercice 2

En utilisant les tableaux de Karnaugh, trouver une expression minimale sous la forme d'une somme de produits des fonctions incomplètement définies suivantes ($d(x) \leftrightarrow \varphi$ dans la case « x »):

$$F_1 = \sum_{W,X,Y,Z} (0,1,3,5,14) + d(8,15)$$

$$F_3 = \prod_{A,B,C,D} (1,5,9,14,15) + d(11)$$

$$F_5 = \sum_{W,X,Y,Z} (3,5,6,7,13) + d(1,2,4,12,15)$$

$$F_2 = \sum_{W,X,Y,Z} (0,1,2,8,11) + d(3,9,15)$$

$$F_4 = \sum_{A,B,C,D} (1,5,6,7,9,13) + d(4,15)$$

XVII. Exercice 3

Trouver une expression minimale sous la forme d'une somme de produits des fonctions suivantes (vous utiliserez la table de Karnaugh) :

$$F_1 = \bar{X} \cdot Z + Y \cdot Y + X \cdot \bar{Y} \cdot Z$$

$$F_2 = \bar{A} \bar{C} \cdot D + \bar{B} \cdot C \cdot D + A \bar{C} \cdot D + B \cdot C \cdot D$$

$$F_3 = W \cdot X \cdot \bar{Z} + W \cdot \bar{X} \cdot Y \cdot Z + X \cdot Z$$

XVIII. Exercice 4

En utilisant des tableaux de Karnaugh à 5 variables, trouver l'expression minimale sous la forme d'une somme de produits des fonctions suivantes :

$$F_1 = \sum_{V,W,X,Y,Z} (5,7,13,15,16,20,25,27,29,31)$$

$$F_2 = \sum_{V,W,X,Y,Z} (1,5,9,13,21,23,29,31,37,45,53,61)$$

Idem en utilisant des tableaux de Karnaugh à 4 variables.

XIX. Détection de validité d'un code en BCD

Soit un groupe de 4 digits composant un nombre codé en BCD : on souhaite créer un circuit dont la sortie « error » = 1 si les 4 digits ne reflètent plus un codage BCD. Proposer le circuit réalisant cette fonction.

XX. L'additionneur

- Rappeler la table de vérité d'une porte « et » et d'une porte « ou exclusif »
- A partir d'une porte « ou exclusive (XOR) » et d'une porte « et (and) », rappeler le circuit du demi-additionneur (sorties « sum » (addition) et sortie « carry » (retenue)).
- Donner la table de vérité de ce circuit demi-additionneur

Considérons un additionneur complet à partir de 2 demi-additionneurs. En entrée nous considérerons deux nombres x et y et une entrée « Cin » correspondant à la retenue de l'étage précédent. Et en sortie nous considérerons « sum » et « carry_out » correspondant à la somme et à la retenue de sortie.

- Donner le circuit global de l'additionneur à partir des 2 demi-additionneurs et d'une porte « ou ».
- Quelle est la différence entre un demi-additionneur et un additionneur ?

XXI. Loi de Morgan

Montrer que les lois de Morgan présentées en cours : $\overline{A \cdot B} = \bar{A} + \bar{B}$ et $\overline{A + B} = \bar{A} \cdot \bar{B}$, est valable pour un nombre quelconque de variables.

XXII. Tableau de Karnaugh 5 variables

Soit la fonction suivante : $F = A \cdot B \cdot C + A \cdot \bar{B} \cdot C \cdot E$. Donner sa forme normale sous une forme de type « E.F1 + /E.F2 ». Sachant que les variables composant F sont A,B,C,D,E. Compléter alors le tableau suivant pour la fonction considérée :

| F | CD | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| AB | 0 | 0 | 1 | 3 | 2 |
| | 1 | 4 | 5 | 7 | 6 |
| | 0 | 12 | 13 | 15 | 14 |
| | 1 | 8 | 9 | 11 | 10 |

Pour E=1

| F | CD | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| AB | 0 | 0 | 1 | 3 | 2 |
| | 1 | 4 | 5 | 7 | 6 |
| | 0 | 12 | 13 | 15 | 14 |
| | 1 | 8 | 9 | 11 | 10 |

Pour E=0

Donner l'expression simplifiée obtenue à partir du tableau.

Poser alors $X = A \cdot C$

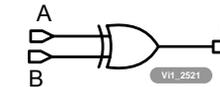
Resimplifier alors l'expression obtenue à partir d'un tableau de Karnaugh.

XXIII. Générateur de parité - nombre impair de 1

Soit la fonction F1 : un générateur de parité impaire est une fonction qui retourne 1 si le nombre de bits à 1 du nombre considéré est impair et 0 sinon.

Redéfinir textuellement cette fonction F1 pour un mot de 4 bits composé des digits ABCD, donner 2 exemples différents de code ABCD et la valeur du bit de parité associé.

Considérons tout d'abord un nombre de 2 digits : AB et le circuit suivant :



- La fonction F est-elle compatible avec la fonction F1 ?
- Proposer alors une fonction et le logigramme de la fonction F1 permettant de traiter le nombre binaire ABCD.



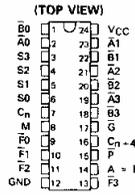
PHYS 246 A Architecture µC - FICHE TD4

I. A propos du composant SN74LS181 – ALU

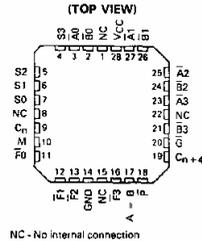
Voici certaines informations données par le constructeurs du composant :

- Full Look-Ahead for High-Speed Operations on Long Words
- Input Clamping Diodes Minimize Transmission-Line Effects
- Darlington Outputs Reduce Turn-Off Time
- Arithmetic Operating Modes:
 - Addition
 - Subtraction
 - Shift Operand A One Position
 - Magnitude Comparison
 - Plus Twelve Other Arithmetic Operations
- Logic Function Modes:
 - Exclusive-OR
 - Comparator
 - AND, NAND, OR, NOR
 - Plus Ten Other Logic Operations

SN64LS181, SN54S181 . . . J OR W PACKAGE
SN74LS181, SN74S181 . . . DW OR N PACKAGE



SN54LS181, SN54S181 . . . FK PACKAGE



description

The 'LS181 and 'S181 are arithmetic logic units (ALU)/function generators that have a complexity of 75 equivalent gates on a monolithic chip. These circuits perform 16 binary arithmetic operations on two 4-bit words as shown in Tables 1 and 2. These operations are selected by the four function-select lines (S0, S1, S2, S3) and include addition, subtraction, decrement, and straight transfer. When performing arithmetic manipulations, the internal carries must be enabled by applying a low-level voltage to the mode control input (M). A full carry look-ahead scheme is made available in these devices for fast, simultaneous carry generation by means of two cascade-outputs (pins 15 and 17) for the four bits in the package. When used in conjunction with the SN54S182 or SN74S182 full carry look-ahead circuits, high-speed arithmetic operations can be performed. The typical addition times shown above illustrate the little additional time required for addition of longer words when full carry look-ahead is employed. The method of cascading 'S182 circuits with these ALUs to provide multi-level full carry look-ahead is illustrated under typical applications data for the 'S182.

If high speed is not of importance, a ripple-carry input (C_N) and a ripple-carry output (C_{N+4}) are available. However, the ripple-carry delay has also been minimized so that arithmetic manipulations for small word lengths can be performed without external circuitry.

recommended operating conditions

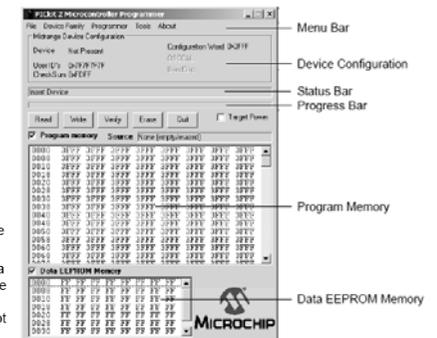
| | SN54S181 | | | SN74S181 | | | UNIT |
|---|----------|-----|-----|----------|-----|------|------|
| | MIN | NOM | MAX | MIN | NOM | MAX | |
| Supply voltage, V _{CC} | 4.5 | 5 | 5.5 | 4.75 | 5 | 5.25 | V |
| High-level output current, I _{OH} (All outputs except A = B) | | | -1 | | | -1 | mA |
| Low-level output current, I _{OL} | | | 20 | | | 20 | mA |
| Operating free-air temperature, T _A | -55 | 125 | 0 | 70 | | 70 | °C |

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ¹ | SN54S181 | | | SN74S181 | | | UNIT |
|--|--|----------|------------------|------|----------|------------------|------|------|
| | | MIN | TYP ² | MAX | MIN | TYP ² | MAX | |
| V _{IH} High-level input voltage | | 2 | | | 2 | | | V |
| V _{IL} Low-level input voltage | | | | 0.8 | | | 0.8 | V |
| V _{IK} Input clamp voltage | V _{CC} = MIN, I _I = -18 mA | | | -1.2 | | | -1.2 | V |
| V _{OH} High-level output voltage, any output except A = B | V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OH} = -1 mA | 2.5 | 3.4 | | 2.7 | 3.4 | | V |
| I _{OH} High-level output current, A = B output only | V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, V _{OH} = 5.5 V | | | 250 | | | 250 | µA |
| V _{OL} Low-level output voltage | V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OL} = 20 mA | | | 0.5 | | | 0.5 | V |
| I _I Input current at maximum input voltage | V _{CC} = MAX, V _I = 5.5 V | | | 1 | | | 1 | mA |
| I _{IH} High-level input current | Mode input | | | 50 | | | 50 | µA |
| | Any A or B input | | | 150 | | | 150 | |
| | Any S input | | | 200 | | | 200 | |
| | Carry input | | | 250 | | | 250 | |
| I _{IL} Low-level input current | Mode input | | | -2 | | | -2 | mA |
| | Any A or B input | | | -6 | | | -6 | |
| | Any S input | | | -8 | | | -8 | |
| | Carry input | | | -10 | | | -10 | |
| I _{OS} Short-circuit output current, any output except A = B ³ | V _{CC} = MAX | -40 | | -100 | -40 | | -100 | mA |
| I _{CC} Supply current | V _{CC} = MAX, T _A = 125°C, W package only | | | 195 | | | | mA |
| | V _{CC} = MAX, See Note 3, All packages | | | 120 | 220 | | 120 | |

1. Souhaitant appliquer un signal numérique sur une des entrées de données de cette ALU, nous souhaitons savoir les caractéristiques des niveaux logiques 0 et 1. Quelles sont-elles ?
2. Nous souhaitons appliquer une tension de 12V entre les pattes 12 et 24. Est-ce concevable ?
3. Il peut être nécessaire d'évaluer la consommation de ce circuit. Quel est le courant maximal utilisé par ce circuit ?
4. Combien d'opérations sont traités par cette ALU pour une opération ? Sur combien de digits travaille cette ALU ? Comment s'effectue le choix entre les différentes opérations possibles ? Quelles sont ces opérations ?

II. A propos du logiciel PicKit2 Programmer - User Guide



1.6.6 Data EEPROM Memory

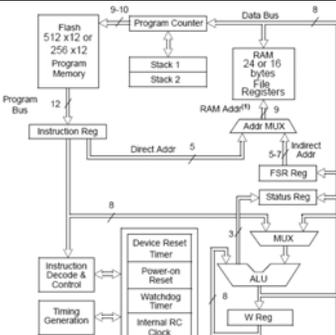
Similar to Program Memory above, program code can be loaded into the PICkit™ 2 Programming Software from *File > Import HEX* or it can be read from the device by clicking on the *Read* button. The origin of the code is displayed in the Source block. The Data EEPROM Memory window displays the program code in hexadecimal. The code cannot be edited in the window.

The check box next to the Data EEPROM Memory window controls whether the data EEPROM memory will be programmed into the device or not. If the box is checked, the code displayed in the Data EEPROM Memory window will be programmed into and used to verify the device. If the box is not checked, the data EEPROM memory will not be programmed and it will not be used to verify the device.

A partir du texte, répondez aux questions suivantes :

1. Ce petit logiciel n'autorise que la programmation du micro contrôleur PIC ?
2. Il est possible de saisir directement le code hexadécimal du programme dans ce logiciel ?
3. Quelle case doit être cochée pour pouvoir envoyer effectivement le code hexadécimal au PIC ?

III. A propos du PIC 10F200

| | |
|---|--|
| <p>3.0 ARCHITECTURAL OVERVIEW</p> <p>The high performance of the PIC10F200/202/204/206 devices can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC10F200/202/204/206 devices use a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architectures where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12 bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (1 µs @ 4 MHz) except for program branches.</p> <p>The table below lists program memory (Flash) and data memory (RAM) for the PIC10F200/202/204/206 devices.</p> | <ol style="list-style-type: none"> 1. Quelle est l'architecture de ce micro contrôleur ? 2. Quelle phrase explicite correctement la spécificité de l'architecture Harvard par rapport à celle de Von Neuman ? |
|  | <ol style="list-style-type: none"> 3. En provenance de la zone mémoire de programme, quelle est la longueur maximale des instructions ? |
| <p>The ALU is 8 bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, one operand is typically the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.</p> <p>The W register is an 8-bit working register used for ALU operations. It is not an addressable register.</p> <p>Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC) and Zero (Z) bits in the Status register. The C and DC bits operate as a borrow and digit borrow out bit, respectively, in subtraction. See the SUBWF and ADDWF instructions for examples.</p> <p>A simplified block diagram is shown in Figure 3-1 and Figure 3-2, with the corresponding device pins described in Table 3-2.</p> | <ol style="list-style-type: none"> 4. quelle est la longueur des opérandes de l'ALU ? 5. Dans le cas d'une addition de 2 opérandes, où sont stockées ceux-ci avant opération ? 6. L'ALU commande le registre d'état. Quelles informations fourni ce dernier dans le cas du 10F200 ? |

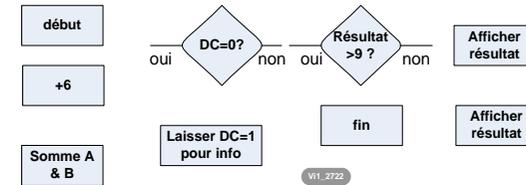
Remarque : C (Carry, report) = le 9eme bit d'une opération

DC (Digit Carry) = DC pour code BCD = report du bit 3 vers le bit 4 (bit4-bit3-bit2-bit1-bit0)

Sur ce que vous avez fait en TD sur la notation BCD et en utilisant l'exemple suivant (l'ALU travaille sur 4 bits)

$$\begin{array}{r}
 . \quad 1 \quad 0 \quad 0 \quad 0 \\
 + \quad . \quad 0 \quad 1 \quad 1 \quad 0 \\
 + \quad . \quad 1 \quad 1 \quad 1 \quad 0 \\
 + \quad . \quad 0 \quad 1 \quad 1 \quad 0 \\
 + \quad . \quad 1 \quad 0 \quad 1 \quad 0 \\
 + \quad . \quad 1 \quad 0 \quad 1 \quad 0 \\
 \hline
 \text{digit DC} \\
 \text{du STATUS}
 \end{array}$$

7. Proposer un logigramme présentant l'algorithme de traitement d'une somme en BCD en utilisant les blocs de base suivants.

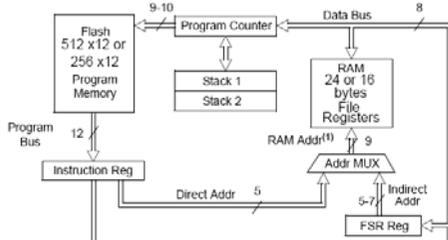


IV. Byte et microcontrôleur PIC 10F200

Voici une partie de la documentation donnée par le constructeur de ce microcontrôleur :

Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12 bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline

| Device | Memory | |
|-----------|----------|--------|
| | Program | Data |
| PIC10F200 | 256 x 12 | 16 x 8 |
| PIC10F202 | 512 x 12 | 24 x 8 |
| PIC10F204 | 256 x 12 | 16 x 8 |
| PIC10F206 | 512 x 12 | 24 x 8 |



Ce processeur comporte une zone « mémoire de programme » et « mémoire de donnée RAM »

- Question 1 : L'une de ces deux zones est une mémoire non volatile.
- Question 2 : Laquelle (expliquez l'utilité de ce caractère permanent de cette zone) ?
- Question 3 : Quelle est la capacité maximale en bytes de la zone « mémoire de données » ?

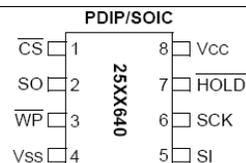
V. A propos du 24AA640 de Microchip - 64K SPI™ Bus Serial EEPROM (21223e.pdf)

2.1 Chip Select (\overline{CS})

A low level on this pin selects the device. A high level deselects the device and forces it into standby mode. However, a programming cycle which is already initiated or in progress will be completed, regardless of the \overline{CS} input signal. If \overline{CS} is brought high during a program cycle, the device will go into standby mode when the programming cycle is complete. When the device is deselected, \overline{SO} goes to the high impedance state, allowing multiple parts to share the same SPI bus. A low to high transition on \overline{CS} after a valid write sequence initiates an internal write cycle. After power-up, a high to low transition on \overline{CS} is required prior to any sequence being initiated.

| Name | PDIP | SOIC | TSSOP | Description |
|-----------------|------|------|-------|--------------------|
| \overline{CS} | 1 | 1 | 3 | Chip Select Input |
| \overline{SO} | 2 | 2 | 4 | Serial Data Output |
| \overline{WP} | 3 | 3 | 5 | Write Protect Pin |
| Vss | 4 | 4 | 6 | Ground |
| SI | 5 | 5 | 7 | Serial Data Input |
| SCK | 6 | 6 | 8 | Serial Clock Input |
| HOLD | 7 | 7 | 1 | Hold Input |
| Vcc | 8 | 8 | 2 | Supply Voltage |

1. Comment faire passer ce composant en mode veille
2. Et comment le rendre opérationnel ?
3. Que se passe-t-il si durant un cycle de fonctionnement, \overline{CS} / passe à l'état haut ?



2.2 Serial Output (SO)

The SO pin is used to transfer data out of the 25XX640. During a read cycle, data is shifted out on this pin after the falling edge of the serial clock.

Description

The Microchip Technology Inc. 25AA640/25LC640 (25XX640) is a 64K bit Serial Electrically Erasable PROM [EEPROM]. The memory is accessed via a simple Serial Peripheral Interface (SPI) compatible serial bus. The bus signals required are a clock input (SCK) plus separate data in (SI) and data out (SO) lines. Access to the device is controlled through a chip select (\overline{CS}) input.

Communication to the device can be paused via the hold pin (HOLD). While the device is paused, transitions on its inputs will be ignored, with the exception of chip select, allowing the host to service higher priority interrupts.

2.4 Serial Input (SI)

The SI pin is used to transfer data into the device. It receives instructions, addresses, and data. Data is latched on the rising edge of the serial clock.

2.5 Serial Clock (SCK)

The SCK is used to synchronize the communication between a master and the 25XX640. Instructions, addresses, or data present on the SI pin are latched on the rising edge of the clock input, while data on the SO pin is updated after the falling edge of the clock input.

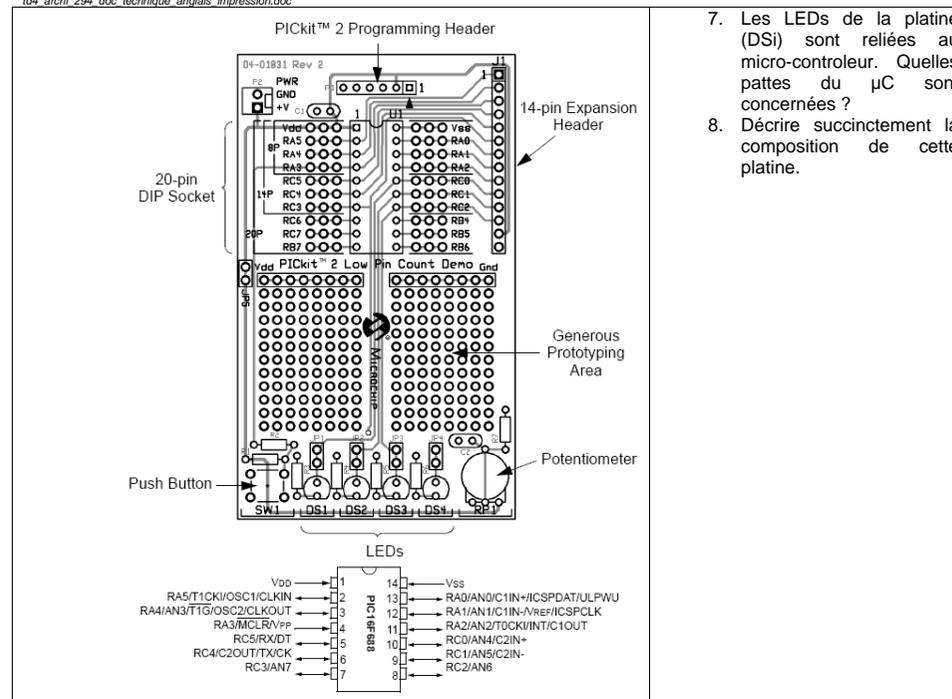
4. La communication avec cette mémoire se fait en utilisant un bus série ou parallèle ?
5. Lors de la lecture d'une information en mémoire, les données sont sorties sur le front montant ou descendant de l'horloge ?
6. Les données entrantes et sortantes passent-elles par la même patte du composant ?

VI. Présentation de la carte PICKit 2 Starter Kit

Ce produit se compose d'un câble USB, d'un programmeur pour PIC, d'une carte de développement munie d'un PIC16F690 et de deux CD.

Le premier CD contient les drivers et les manuels d'utilisation. Le second contient les logiciels pour programmer les PIC, tel que MPLAB, et des exemples codés en Assembleur et en C pour se familiariser avec ces langages. Il est à noter que le contenu des deux CD est téléchargeable gratuitement sur le site de Microchip.

La carte de développement est appelée : Low Pin Count Demo Board. Son schéma électrique est présenté ci-dessous :



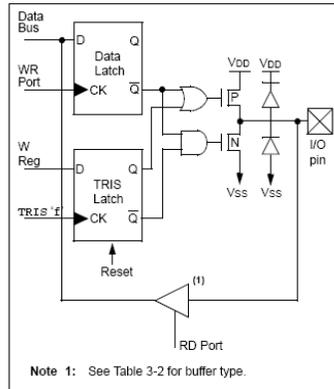
7. Les LEDs de la platine (DSi) sont reliées au micro-contrôleur. Quelles pattes du μC sont concernées ?
8. Décrire succinctement la composition de cette platine.

VII. Configuration des entrées sorties

5.3 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-2. All port pins, except GP3 which is input only, may be used for both input and output operations. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., MOVF GPIO, W). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit in TRIS must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin (except GP3) can be programmed individually as input or output.

FIGURE 5-1: PIC10F200/202/204/206 EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



1. La majorité des pattes du µC est-elle en entrée ou sortie ?
2. quel est le nom du registre qui permet de paramétrer le sens Input/output des pattes du µC ?
3. Nous souhaitons paramétrer la patte en sortie. Quelle valeur doit on attribuer au bit « direction control bit » du registre concerné ?

| Address | Bank 0 | Bank 1 | Address |
|-----------|---------------|------------|-----------|
| 00h | INCF | ← | 80h |
| 01h | TMR0 | OPTION_REG | 81h |
| 02h | PCL | ← | 82h |
| 03h | STATUS | ← | 83h |
| 04h | FSR | ← | 84h |
| 05h | PORTA | TRISA | 85h |
| 06h | PORTB | TRISB | 86h |
| 07h | Unimplemented | ← | 87h |
| 08h | EEDATA | EEDON1 | 88h |
| 09h | EEDADR | EEDON2 | 89h |
| 0Ah | PCLATH | ← | 8Ah |
| 0Bh | INTCON | ← | 8Bh |
| 0Ch - 4Fh | GPR | ← | 8Ch - CFh |

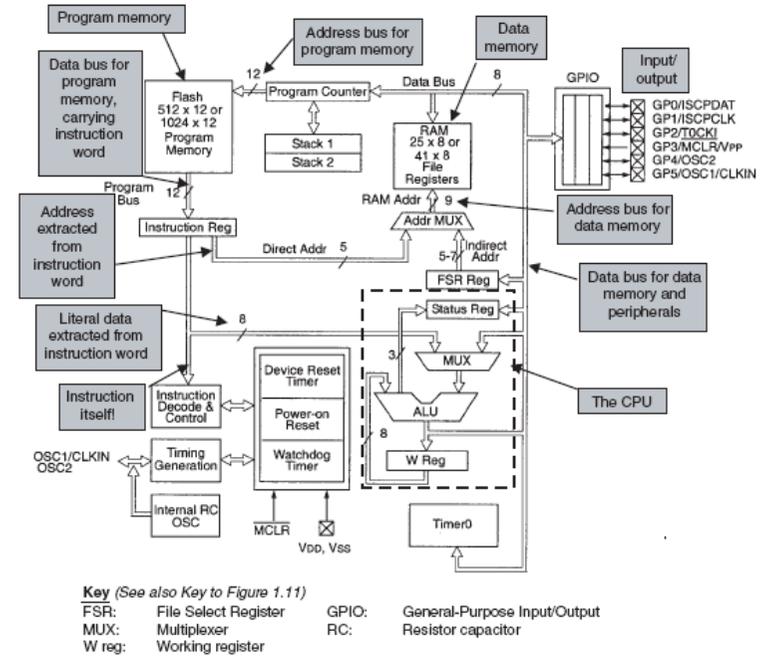


Figure 1.13 PIC12F508/509 block diagram (supplementary labels in shaded boxes added by the author)

VIII. Famille des Pic

Comparer les architectures internes de ces deux microcontrôleurs en répondant aux questions :

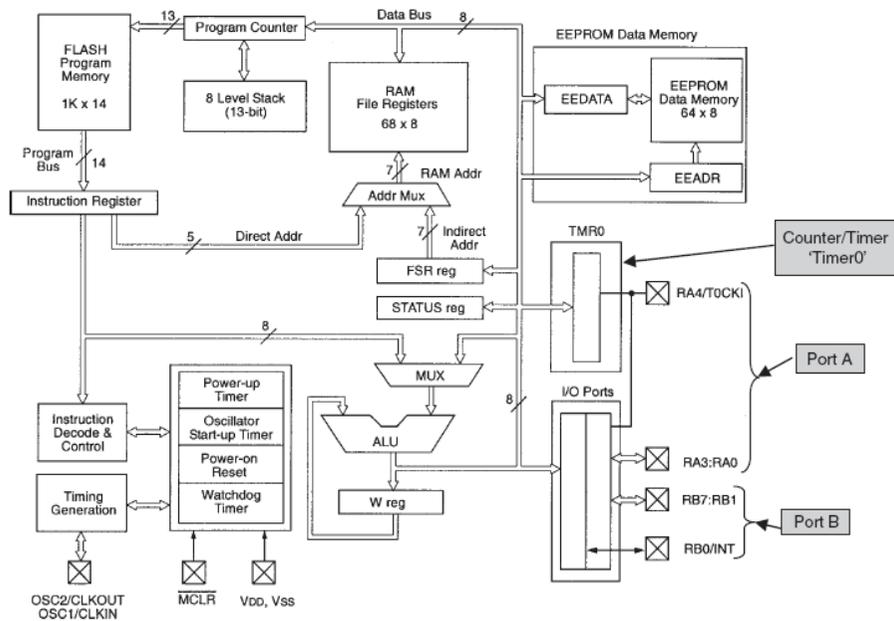


Figure 2.2 Block diagram of the 16F84A (supplementary labels in shaded boxes added by the author)

Architecturally there is clear similarity between the 12F508 and the 16F84A. In fact, the former is a direct subset of the 'F84A, with near identical CPU, memory, bus structure and counter/timer (TMR0) peripheral. Notice first, however, that the address bus sizes have been increased, to meet the needs of the whole PIC 16 Series family. As a smaller member of that family, the 'F84A doesn't fully exploit all these developments. The program address bus is now 13-bit and the instruction word size is 14-bit. Therefore, 2^{13} (i.e. 8192) memory locations *could* be addressed. Program memory size, at 1K, is however only one-eighth of this. The larger bus size will, however, be useful in the larger 16 Series devices, as can be seen in the program memory size of the 16F876A and 16F877A (Table 2.1). RAM size has crept up cautiously to 68 locations and the Stack to eight locations.

A number of important new additions have appeared. The addition of an EEPROM memory gives the valuable capability of being able to store data values even when the chip is powered down. There are now two digital input/output ports. These are Port A, with five pins, and Port B, with eight. Importantly, there is the addition of an interrupt capability (which we explore in detail in Chapter 6). This can be seen externally on pin 6, where bit 0 of Port B is shared with the external interrupt input. We will also see that there are three further internal interrupt sources, generated by the peripherals.

1. quelles sont les différences sur le bus d'adresse de la mémoire de programme ?
2. Pour le pic16F, de combien de port I/Os dispose-t-il ?
3. comparer la taille de la zone mémoire de programme de ces deux microcontrôleurs ?
4. y a-t-il une possibilité de stocker des données de façon non volatile dans un de ces deux composants ?
5. localiser la patte du composant 16F qui détecte les interruptions externes ?
6. toujours pour ce même composant, y a-t-il des possibilités de détection d'interruptions internes ?

PHYS 246 A Architecture µC – FICHE **TD5**



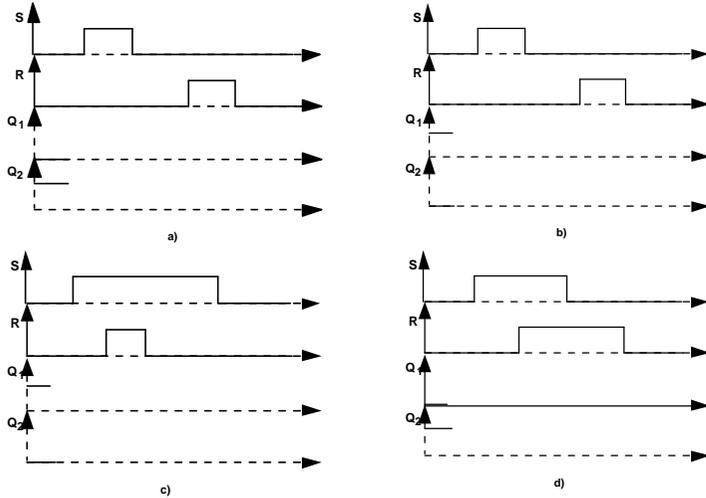
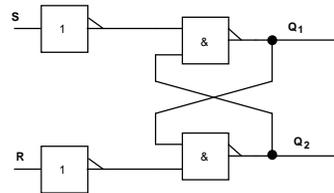
I. Exercice N°1 : l'opérateur NAND

A partir de portes NAND à 2 entrées, réaliser les fonctions logiques suivantes :

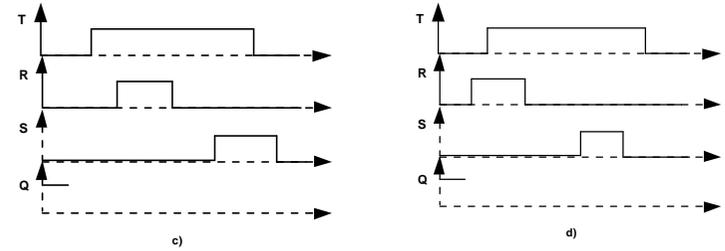
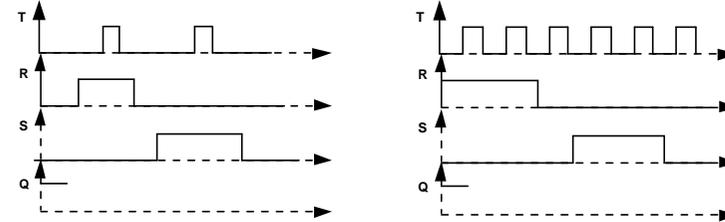
- a) /A
- b) A.B
- c) A+B

II. Exercice N°2 : Les bascules

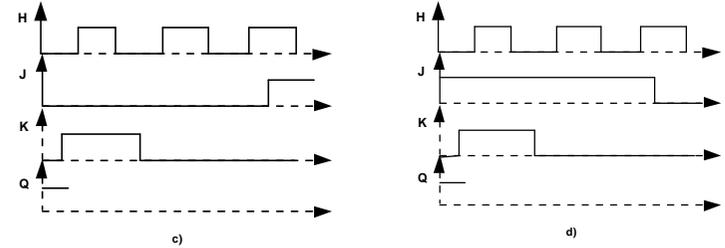
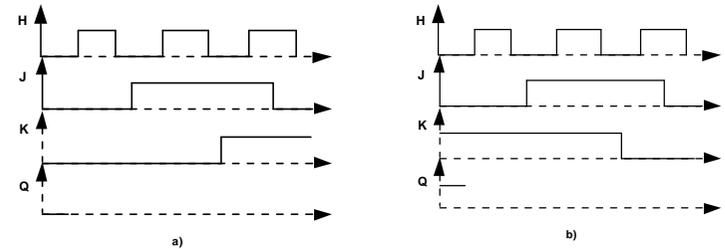
- 1) En considérant le logigramme ci-dessous, donner le type de bascule en présence et les portes de bases constituant cette bascule
- 2) Compléter les chronogrammes suivants pour la bascule RS ci-dessous:



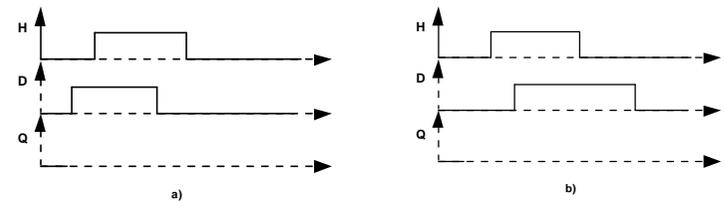
- 3) Mêmes exercices avec une bascule RST (synchronisée sur le niveau haut de T).

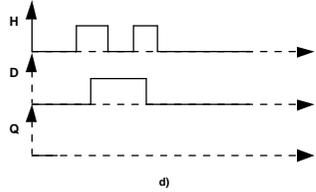
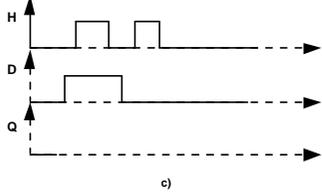


- 4) Mêmes exercices avec une bascule JK synchronisée par un front positif.

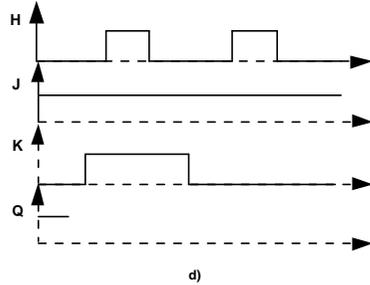
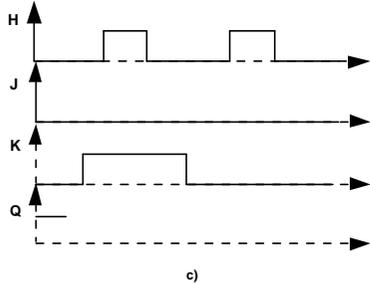
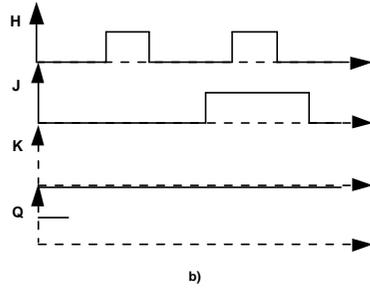
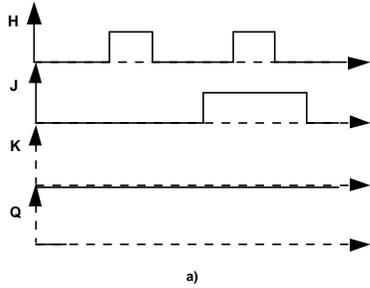


- 5) Mêmes exercices avec une bascule D latch.



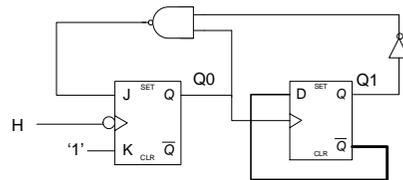


6) Mêmes exercices avec une bascule JK maître-esclave.



III. Exercice N°3 : Chronogramme et bascules

En considérant le circuit ci contre, complétez le chronogramme des états (Q_0, Q_1) ci-dessous, celui-ci étant supposé à $(00)_2$ au départ. Vous tiendrez compte des retards induits par les bascules. Ces retards sont identiques pour les deux bascules et valent Δt (passage de l'état haut vers bas et bas vers haut)



IV. Exercice N°4 : bascules et cycle

Analyser le cycle réalisé par le circuit présenté en ci contre, celui-ci étant supposé à $(0000)_2$ initialement. Le comportement est il cyclique ?

