

(XIV) PRATIQUE DE LA MESURE

COMPTEURS

ET FREQUENCEMETRES

Après avoir étudié pendant plusieurs mois l'oscilloscope et son utilisation, sans avoir sans doute épuisé le sujet, nous allons tout de même passer à d'autres types de mesure. Nous parlerons dans nos prochains articles des compteurs électroniques, dont l'apparition ne remonte qu'à une bonne dizaine d'années, surtout dans le domaine des amateurs.

En fait, le compteur numérique est le fils aîné de l'électronique logique dont il est bon de faire un rapide rappel historique, marqué essentiellement

par les progrès incessants de la technologie permettant de mettre à la disposition des concepteurs de systèmes des composants de plus en plus performants.

Ces composants sont d'ailleurs essentiellement des CIRCUITS INTEGRES logiques, dont l'apparition remonte à peu près aux années 1966-1967, à partir desquelles ces circuits commencèrent à être disponibles à prix abordables. Successivement apparurent puis disparurent un certain nombre de « familles » de circuits logiques.

1. La RTL (Resistor Transistor Logic)

Il s'agit d'une famille de circuits dans laquelle les éléments de base, les portes, sont réalisées en utilisant des transistors et des résistances. La figure 1 nous montre ainsi la réalisation d'une porte NAND (ou Non ET). Les résistances R_1 à R_4 sont calculées pour que le transistor T ne soit saturé que si les deux entrées A et B sont au potentiel positif (niveau 1), égal à +E. Si l'une des entrées est à 0, T reste bloqué. Ceci nous conduit donc à la table de vérité ci-dessous, valable pour toutes les portes NAND, quelle que soit leur technologie.

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

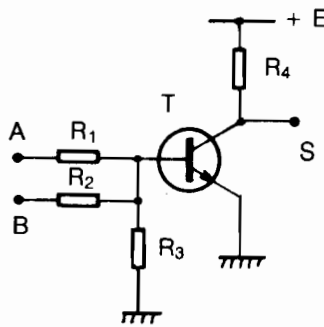


FIGURE 1. - NAND en RTL.

Les circuits RTL étaient lents, leur fabrication très critique car la valeur des résistances de base difficile à réaliser avec précision, lorsque le nombre des entrées dépassait 3. Ces circuits sont totalement abandonnés.

2. La DTL (Diode Transistor Logic)

Les résistances de la figure 1 sont remplacées par des diodes. Voir la fi-

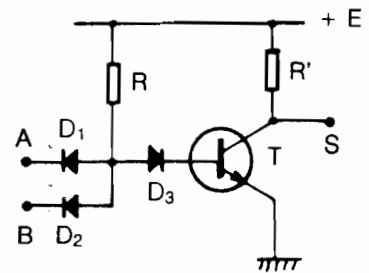


FIGURE 2. - NAND en DTL.

gure 2, correspondant toujours à une porte NAND.

Entrées en l'air, T est saturé par la résistance R et S est au niveau 0.

Entrées au +E, il en est de même, les diodes ne conduisant pas lorsqu'elles ont leur cathode côté +.

Mais, dès que l'une des entrées est reliée à la masse, soit au niveau 0, cette diode conduit et porte le potentiel de D_3 à une valeur voisine de 0, ce qui bloque le transistor T. La diode D_3 renforce ce blocage. La sortie S passe ainsi à 1. La table de vérité précédente est bien encore obtenue.

Le progrès majeur de la DTL sur la RTL est que le nombre des entrées peut être aussi grand que désiré : la saturation de T ne dépend que de R et pour son blocage, il suffit d'une diode à la masse, plusieurs en même temps ne faisant pas mieux.

3. La TTL (Transistor Transistor Logic)

Les diodes de la figure 2 sont remplacées par un transistor multi-émetteur. Si les entrées sont en l'air, la pseudo-diode de la jonction base-collecteur de T' conduit et sature T, la sortie S est à 0.

Si les entrées sont à 1, il en est de même.

Si l'une des entrées, c'est-à-dire un des émetteurs de T' est à la masse, ce transistor se met à conduire normalement, entre collecteur et émetteur : il relie donc la base de T à un potentiel voisin de la masse : T se bloque et fait passer la sortie S au niveau 1.

Au niveau statique, il n'y a pas de différence de fonctionnement entre la DTL et RTL. Par contre, cette différence est capitale en régime dynamique.

La mauvaise performance au niveau vitesse de la DTL s'explique par le fait que, au blocage de T, les charges des porteurs minoritaires de base amenés par l'état de saturation ne peuvent pas s'écouler hors de cette base, la diode D₃ étant bloquée. Ils doivent s'évacuer « naturellement » dans le transistor et cela prend « un certain temps »...

Par contre, en TTL, au blocage, le transistor T' est conducteur. Ces porteurs peuvent ainsi s'évacuer très rapidement à travers T'. D'où un gain de temps significatif au blocage de T. La vitesse de commutation possible avec la TTL tombe à quelque 12 ns.

La famille TTL s'est vite enrichie de plusieurs « enfants » :

- La HTTL : plus rapide encore que la TTL normale, avec un temps de commutation de 8 ns environ, et une consommation un peu supérieure, cette variété est peu courante.

- La STTL : c'est la plus rapide de toutes : 3 ns. Mais consommation encore plus élevée. La vitesse est obtenue par l'utilisation de transistors de type Schottky.

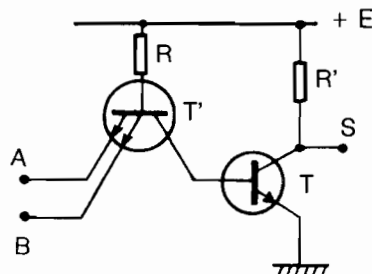


FIGURE 3. - NAND en TTL.

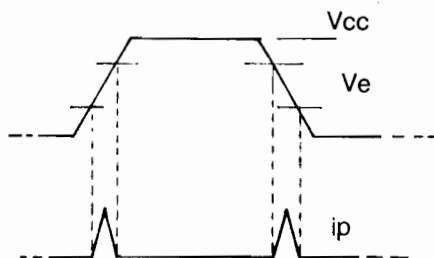


FIGURE 5. - Consommation de courant pendant les transitions.

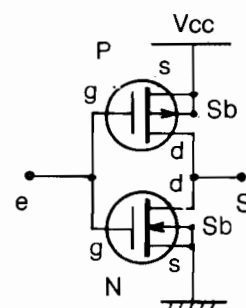


FIGURE 4. - Inverseur C-MOS.

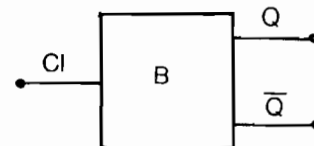


FIGURE 6 Basculeur élémentaire.

- La LTTL : version à faible consommation de la TTL classique, celle-ci est à peu près réduite dans un facteur de 10. La vitesse tombe à une trentaine de nanosecondes.

- La LSTTL : c'est la version la plus récente. Cette variété concilie à la fois vitesse et consommation : L pour faible courant et S pour vitesse. Cette série est encore la plus utilisée de nos jours, en particulier en informatique.

Si les familles ci-dessus sont toutes à base de transistors bipolaires classiques, ce qui explique leur consommation toujours assez élevée, même au repos, la famille qui suit utilise une tout autre technologie : celle des transistors à effet de champ (MOS), de types complémentaires (C), avec grilles de commandes isolées.

C'est la fameuse C-MOS.

La figure 4 montre le schéma de base de la paire complémentaire utilisée. C'est la structure de base des circuits logiques C-MOS. On y voit deux transistors MOS, le supérieur de type P, l'inférieur de type N.

Si l'entrée est portée à Vcc, le transistor N conduit et relie la sortie S à la masse : S = 0, si E = 1.

Si l'entrée est portée à 0 V, c'est le transistor P qui devient conducteur, ce qui relie S à Vcc : S = 1 si E = 0.

Le montage obtenu est donc un inverseur. Notons quelques points très importants :

- Au repos, la consommation est quasi nulle : il n'y a pratiquement aucun courant d'entrée (grilles isolées) et pas de passage entre Vcc et masse puisque l'un des deux transistors est toujours bloqué.

- Les niveaux de sortie sont pratiquement égaux aux tensions d'alimentation, les deux transistors ayant des résistances de passage très faibles. Ce n'était pas le cas dans les autres familles.

- Pratiquement, la consommation des C-MOS n'apparaît qu'en régime dynamique, au moment de la transition de la tension de grille de 0 à 1. En effet, lorsque cette tension est de l'ordre de 1/2 Vcc, les deux transistors tendent à conduire en même temps. Voir figure 5. Plus il y a de commutations et plus il y a de consommation : celle-ci est donc proportionnelle à la fréquence des signaux transmis.

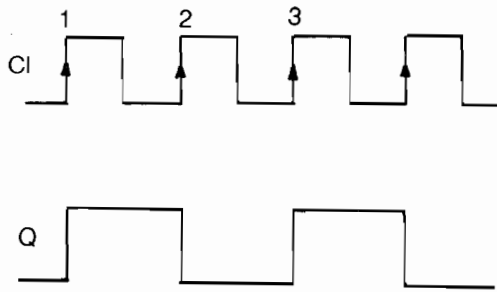


FIGURE 7. – Un basculeur divisé par 2.

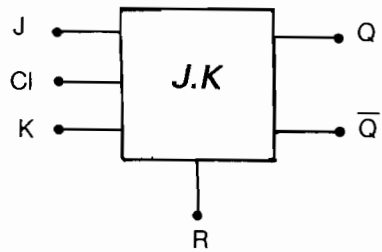


FIGURE 8. – Le basculeur JK.

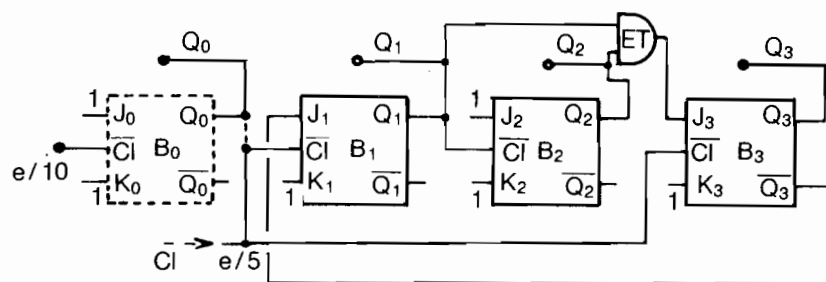


FIGURE 9. – Une décade comprend un diviseur par 5 (B₁, B₂, B₃) précédé d'un diviseur par 2 (B₀).

– Limitée en fréquence, pendant quelques années (vitesse de l'ordre de 100 à 200 ns), on commence depuis peu à trouver de la C-MOS haute vitesse. C'est la HC-MOS, aussi rapide que la LSTTL mais bien moins gourmande. Cette nouvelle famille remplacera sans aucun doute toutes les autres, dans les années qui viennent. En effet, si avec une fréquence élevée, la différence de consommation entre LSTTL et HC-MOS n'est plus très importante, en revanche, cette HC-MOS conserve ses autres avantages : entrées à très haute impédance et consommation quasi nulle, ce qui permet, entre autres, de connecter un nombre quelconque d'entrées sur une sortie unique (ce n'est pas le cas en TTL où les problèmes d'entrance et de sortance sont délicats) et par ailleurs la tension des sorties calibrées aux potentiels de l'alimentation.

Ce petit tour d'horizon étant fait, nous pourrions parler des montages les plus classiques de l'électronique numérique, à savoir les compteurs.

Ce sont d'ailleurs des montages en principe très simples, puisqu'il doivent

uniquement dénombrer des impulsions.

Pour y parvenir, nous utiliserons essentiellement des systèmes réalisés avec des associations de portes NAND : les basculeurs, lesquels, comme leur nom l'indique, peuvent prendre un état (0) ou un autre (1), en fonction des impulsions qu'ils reçoivent.

Le basculeur le plus simple est représenté en figure 6. Il dispose d'une entrée dite d'horloge (Clock) et de deux sorties complémentaires Q et Q-bar. Initialement au repos, avec Q = 0 et Q-bar = 1, la première impulsion le fait passer au travail : Q = 1 et Q-bar = 0. La seconde impulsion le ramène au repos, la troisième au travail...

Comme le montre la figure 7, le basculeur s'avère être un diviseur par 2. En associant plusieurs montages similaires en cascade, il est donc aisé de réaliser des diviseurs par 4, 8, 16, 32.

Mais, vous le savez, les humains comptent par 10. Ils utilisent le système décimal. Et malheureusement, le nombre 10 ne fait pas partie de la série ci-dessus. Dix n'est pas une puissance de

2 ! Pour réaliser un diviseur par 10, il nous faut d'abord perfectionner le basculeur élémentaire de la figure 6. Nous allons lui adjoindre deux entrées supplémentaires de contrôle : J et K. Nous lui donnerons aussi une entrée de remise à 0 (RAZ). Voir figure 8.

Quel est le rôle de J et K ?

Supposons le basculeur remis à 0, par R par exemple.

– Si J et K sont à 1, le basculeur fonctionne exactement comme avant, il divise par 2 le nombre des impulsions d'horloge.

– Si J et K sont à 0, le basculeur est bloqué, insensible aux impulsions d'entrée. Il reste dans l'état où il était au moment de cette action.

– Si J = 1 et K = 0, le basculeur bascule une fois en passant au travail, puis il se bloque dans cet état : Q = 1 et Q-bar = 0.

– Si J = 0 et K = 1, c'est l'inverse, il bascule une fois au repos et y reste.

Notre basculeur ainsi amélioré, passons à la réalisation d'un diviseur par 5 (voir figure 9).

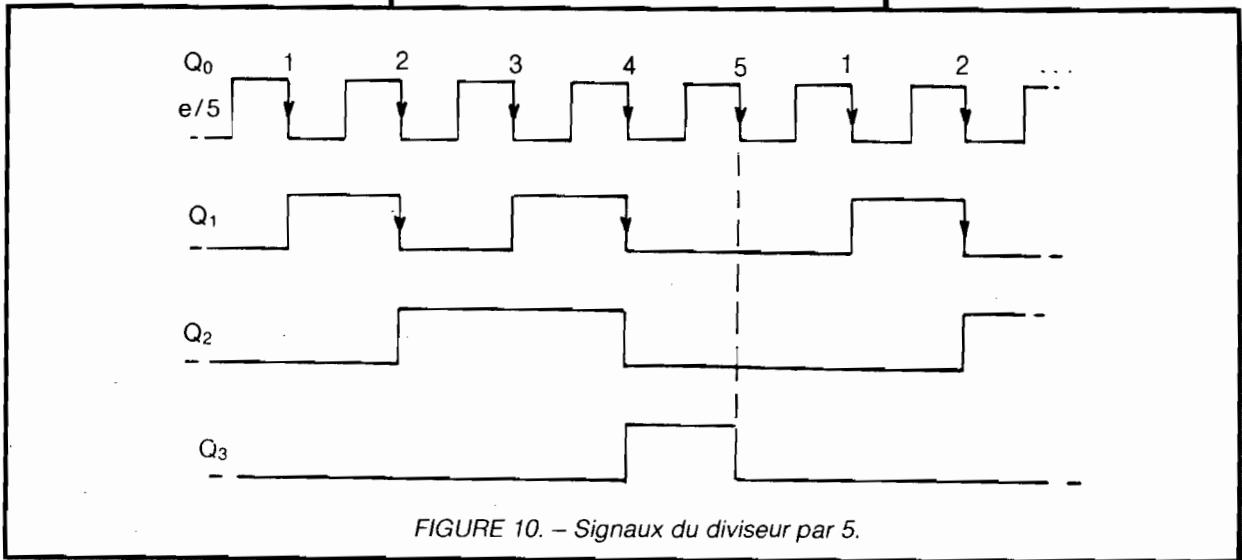


FIGURE 10. - Signaux du diviseur par 5.

Observons tout d'abord que B_2 , avec J_2 et K_2 à 1, divise toujours par 2. B_1 divise par 2 si B_3 est au repos, puisque $J_1 = \bar{Q}_3 = 1$. Enfin B_3 bascule selon l'état de J_3 , qui dépend de Q_1 et de Q_2 .

Imaginons le système au repos. Tous les Q sont à 0 et tous les \bar{Q} à 1. B_1 et B_3 reçoivent le signal d'horloge direct. B_1 peut basculer sur les fronts descendants, mais B_3 ne le peut pas ($J_3 = 0$). Les sorties se positionnent donc suivant le diagramme de la figure 10, Q_2 basculant sur les fronts descendants de Q_1 .

Après la troisième impulsion, on a $Q_1 = Q_2 = 1$, donnant, via la porte ET, $J_3 = 1$. A l'impulsion suivante, donc à la quatrième, B_3 bascule et de ce fait

bloque B_1 , d'où B_2 . A la cinquième impulsion, B_3 revient au repos et... on retrouve alors le système dans son état initial avec $Q_1 = Q_2 = Q_3 = 0$.

Toutes les cinq impulsions, le système recycle. Il compte par 5. Il sort une impulsion de Q_3 toutes les cinq impulsions appliquées à l'entrée.

Vous l'avez deviné, il suffit d'ajouter devant cet ensemble un quatrième diviseur B_0 , divisant normalement par 2, pour obtenir un diviseur par 10, puisque 2×5 font 10. Les quatre basculeurs constituent une DECADE.

Nous venons d'étudier l'élément de base des compteurs décimaux. Les décades se trouvent complètes dans des circuits intégrés : le 7490, le 74196 en TTL ou LSTTL. Il existe même des dé-

cadés doubles : le 74390. Des décades existent aussi en C-MOS, bien entendu.

Pour fabriquer un véritable compteur, il suffit de connecter en cascade plusieurs décades : six, par exemple, pour compter de 1 à 999999 ! Toutes les entrées R sont reliées en parallèle. Il suffit de porter R à 1 pour remettre le compteur entier à 0.

Mais n'oublions pas que notre décade de base est tout de même binaire. Son état se lit sur les sorties Q_0 à Q_3 . (Remarquons que CI de la figure 10 correspond à Q_0 du quatrième basculeur.) La conversion binaire/décimal des états montre bien le comptage par 10, de 0 à 9.

Malheureusement, ces états binaires ne sont pas directement lisibles par

Q_3	Q_2	Q_1	Q_0	Etat
0	0	0	0	0
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

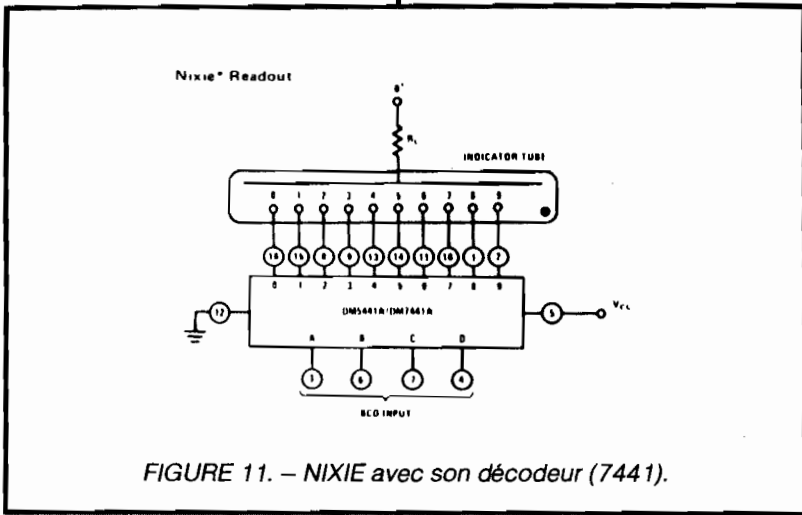


FIGURE 11. – NIXIE avec son décodeur (7441).

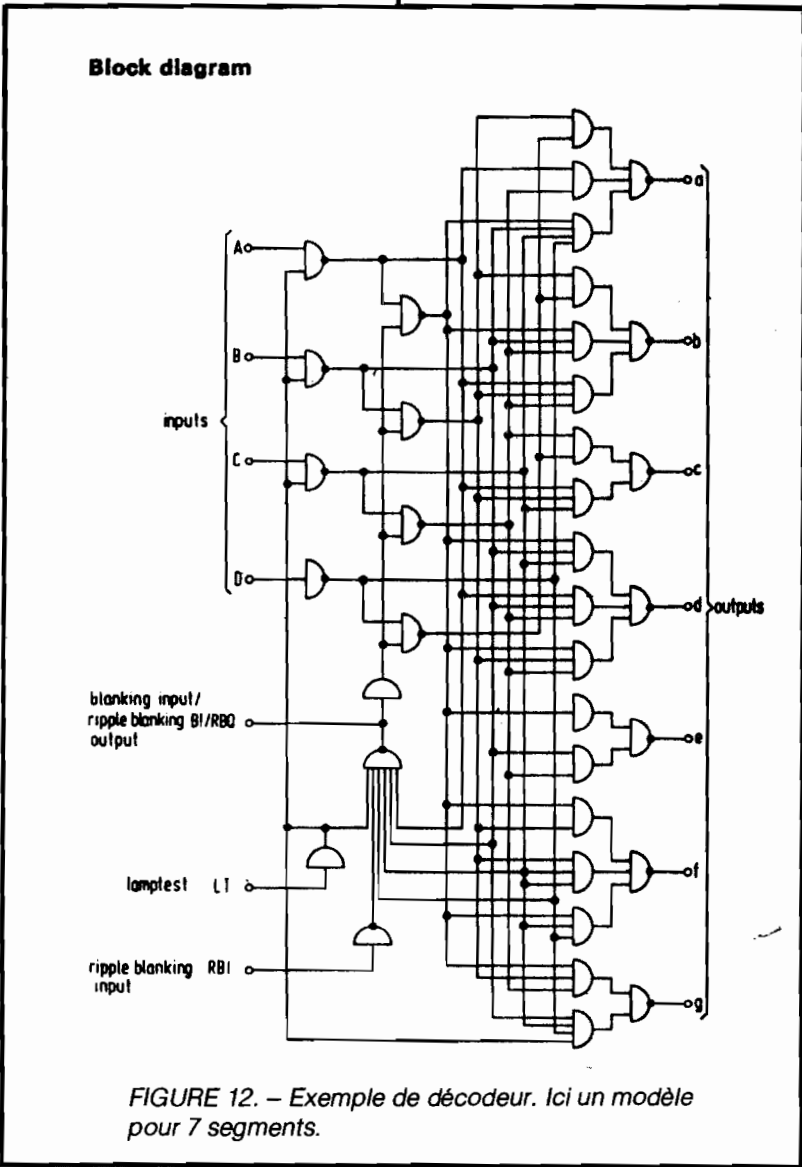


FIGURE 12. – Exemple de décodeur. Ici un modèle pour 7 segments.

« l'homme » qui compte et voit en décimal. Il faut donc intercaler, entre chacune des décades et l'opérateur, un « traducteur » – on dit un « décodeur » – binaire/décimal.

Les premiers décodeurs réalisés étaient destinés à des afficheurs NIXIES, tubes électroniques à atmosphère au néon et qui possédaient autant d'électrodes que de signes à afficher, chaque électrode ayant la forme de son signe. Ces électrodes, en fil très fin étant placées les unes derrière les autres. Une anode commune est portée à haute tension (+ 200 V). Dès que l'une des électrodes (cathode) est reliée à la masse, elle s'ionise et devient lumineuse (voir figure 11). Pour de tels afficheurs, le décodeur doit avoir 10 sorties, une par électrode et 4 entrées, une par Q de la décade. Un réseau complexe de portes NAND intégrées assure la conversion nécessaire (voir figure 12).

Actuellement, la solution NIXIE est abandonnée. Elle avait l'inconvénient de la tension élevée nécessaire, et surtout de l'affichage des divers symboles dans des plans différents, ce qui rendait impossible une observation oblique. Nous en sommes maintenant à l'affichage « 7 segments », que chacun connaît pour l'avoir observé dans tous les appareils à affichage : montres et horloges en particulier. Chaque chiffre est dessiné avec ces 7 segments. Ce n'est pas très joli, mais on y est habitué (voir figure 13).

Remarquer la forme désagréable du « 6 » et du « 9 ». En fait, cette forme, qui supprime le segment « a » pour le 6 et « d » pour le 9, a été étudiée pour réduire au minimum le nombre des lectures erronées, en cas de défaillance d'un segment. La figure 14 vous montre ces erreurs possibles dans les deux cas.

Les circuits 7447, en TTL, et 4511 en C-MOS suivent un tel codage. Par contre, pour ceux que cette forme des 6 et 9 rend malades, le 74247 est la solution à adopter, en TTL, tandis que le 4513 sera retenu en C-MOS.

A noter que les décodeurs ci-dessus ne sont correctement utilisables qu'en décimal, à partir d'une décade, tandis que le C-MOS MC4495-1 permet de plus l'affichage des « chiffres » hexadécimaux A, B, C, D, E et F. Ceci est fort utile pour les amateurs d'informatique.

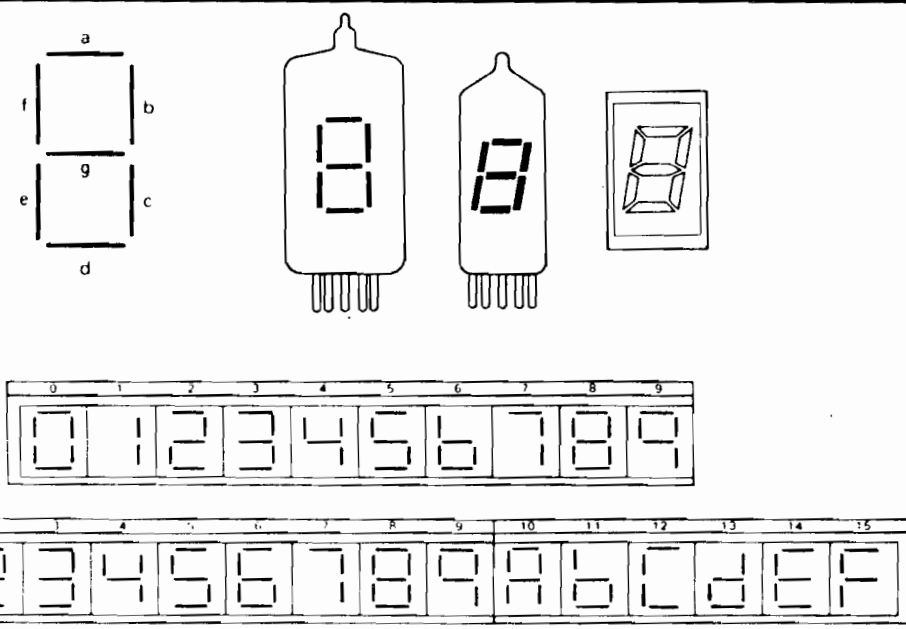


FIGURE 13. - Afficheurs 7 segments. Certains décodeurs ne sont que de 0 à 9 (4511), d'autres jusque F (4495).

SEVEN SEGMENT DECODER/DRIVER WITH NUMERIC ONE ON THE RIGHT HAND SIDE AND TAILS ON NUMERIC SIX AND NINE

0	1	2	3	4	5	6	7	8	9	DECODE FORMAT
0	1	2	3	4	5	6	7	8	9	SEGMENT (a) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (b) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (c) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (d) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (e) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (f) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (g) FAILS

9317 DECODER-DRIVER WITH NUMERIC ONE ON THE LEFT HAND SIDE AND NO TAILS ON NUMERIC SIX AND NINE

0	1	2	3	4	5	6	7	8	9	DECODE FORMAT
0	1	2	3	4	5	6	7	8	9	SEGMENT (a) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (b) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (c) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (d) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (e) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (f) FAILS
0	1	2	3	4	5	6	7	8	9	SEGMENT (g) FAILS

FIGURE 14. - Etude systématique de l'erreur sur coupure d'un segment !

A gauche : avec « queues » sur 6 et 9 et 1 à droite : 7 cas d'erreurs.

A droite : sans « queues » et 1 à gauche : 2 erreurs, les anomalies A, B et C étant dépis-tables par un observateur attentif.

... Et voilà pourquoi ils font ainsi le « 6 » et le « 9 » !

En conclusion

Nos décades et nos décodeurs, associés à des afficheurs à 7 segments, vont nous permettre la réalisation de

compteurs électroniques exactement semblables aux compteurs mécaniques (compteurs kilométriques des voitures, par exemple). A cette différence énorme près : leur vitesse de comptage, pouvant atteindre en LSTTL ou

en HC-MOS les 50 MHz. Compter 50 millions d'impulsions par seconde. Avouez que c'est assez déconcertant quand on y réfléchit un peu. Et pourtant, on fait mieux, nous le verrons plus tard.

F. THOBOIS