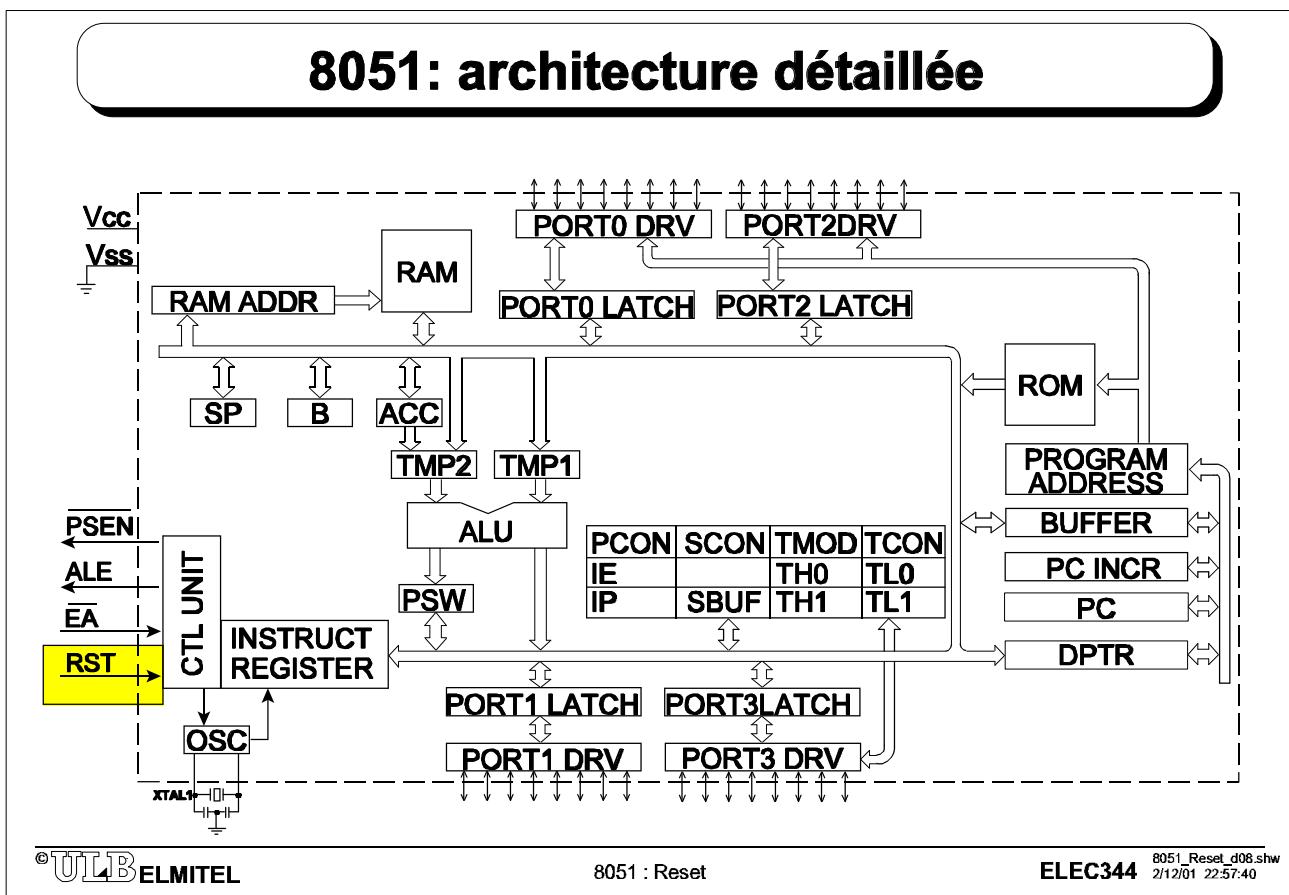


ÉLECTRONIQUE

micro-contrôleur 8051 : RESET

8051: architecture détaillée



©ULB ELMITEL

8051 : Reset

ELEC344 8051_Reset_d08.shw
2/12/01 22:57:40

3

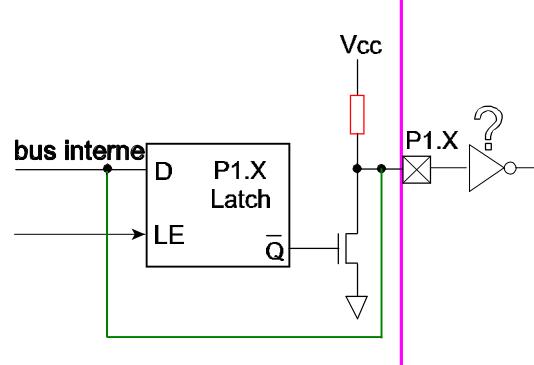
Un schéma bloc plus détaillé nous montre l'architecture interne du 8051 de base. Nous n'allons pas décrire tout le schéma d'un coup, mais bien le parcourir pas-à-pas en commençant par le début, c'est-à-dire l'allumage du système.

Lors de la mise sous tension, la ligne d'entrée RESET doit être mise dans son état actif le plus tôt possible.

Cette entrée agit principalement sur le moteur logique du processeur : l'**unité de contrôle** dite aussi **unité de commande** ou "control unit" à gauche de la figure. Celle-ci est une machine d'état (logique séquentielle) responsable du déroulement de toutes les étapes, à tous les stades de fonctionnement du processeur. Un chapitre spécial y sera consacré.

Buts du RESET à l'allumage

- ▶ tant que l'oscillateur n'a pas démarré, empêcher le fonctionnement
 - ◆ du µP
 - ◆ de tous les périphériques
- ▶ mettre le système dans un état connu
 - ◆ séquenceur de fonctionnement
 - ◆ registres essentiels
 - ◆ ports de sortie
 - ◆ périphériques
- ▶ le RESET doit être distribué simultanément à tous les boîtiers



le RESET du µP doit mettre tout le système dont il dépend dans un état sûr

Par analogie avec l'entrée RESET des bistables, le RESET des processeurs met ceux-ci dans un état de repos connu. Elle est active par niveau : l'état de RESET se maintient tant que l'entrée est active.

Le RESET doit être actif pendant un temps supérieur au démarrage de l'oscillateur d'horloge c'est-à-dire un délai de quelques ms à quelques dizaines de ms.

Après cela, le processeur peut commencer à exécuter son programme en commençant par toutes les initialisations nécessaires.

L'état de toutes les sorties du processeur et des périphériques lors du RESET doit absolument correspondre à un état sûr du système.

La figure illustre ici un bit d'un port d'entrée-sortie quasi-bidirectionnel d'un micro-contrôleur (voir le chapitre sur les entrées-sorties parallèles pour les détails). Lors du RESET, le port est mis en entrée avec une résistance de "pull-up" qui fixe le niveau par défaut à HI. Supposons que ce bit doive-t-être une sortie. En attendant que la routine d'initialisation du micro-contrôleur ait fixé le niveau du bit de sortie en écrivant dans le latch, il faut que le niveau HI par défaut corresponde à un état acceptable du système. Si ce n'est pas le cas, il faut ajouter un inverseur à la sortie.

Buts du RESET en fonctionnement

- ▶ à cause d'un plantage
 - ◆ d'origine software "bug"
 - ◆ d'origine hardware : parasite, particule d'énergie suffisante
- ▶ activé par
 - ◆ l'opérateur
 - ◆ le "watchdog" interne
- ▶ remet le µP dans son état de départ
- ▶ intérêt d'une mémorisation régulière de l'état pendant le fonctionnement
 - ◆ boîte noire
 - ◆ rattrapage au vol

En cas de dérapage du programme, pour une cause logicielle (bogue ou "bug") ou pour une cause matérielle (comme un parasite), le RESET servira à réinitialiser le processeur dans le même état qu'au démarrage, pour reprendre l'exécution du programme sur une base connue et propre.

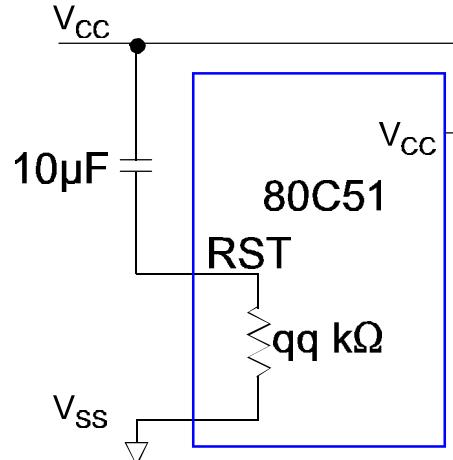
Il sera provoqué :

- soit par l'opérateur qui se rend compte du plantage
- soit par un "watchdog" interne. Celui-ci est un timer qui active le RESET s'il déborde. Il doit donc être remis à zéro avant son débordement par le programme en fonctionnement normal. En cas de plantage, on espère que la remise à zéro finira par ne pas se produire dans le délai prévu.

Il peut être intéressant de sauver certaines variables d'état du système de manière non-volatile à des fins de diagnostic (boîte noire), ou pour faciliter un "rattrapage au vol", c'est-à-dire reprendre le fonctionnement le plus proche possible du dernier état avant le RESET.

Circuit de Reset du 8051 (simple)

- ! 8051 = cas particulier
 - ♦ reset actif à l'état haut
 - ♦ nécessite
 - alimentation avec temps de montée court (<1ms)
 - démarrage de l'oscillateur <10ms
- ! pattes de sortie dans un état quelconque pendant tout le démarrage de l'oscillateur + 24 cycles



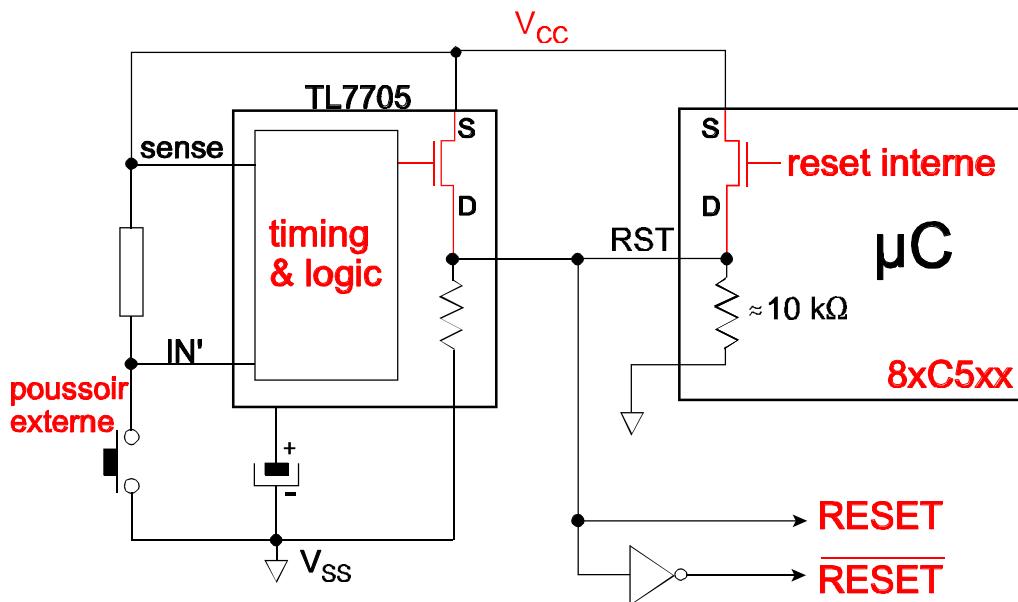
Le 8051 est un bon exemple de micro-contrôleur, mais plutôt un mauvais exemple de RESET.

Il existe bien un circuit très simple qui maintient le RESET actif grâce à un délai créé par une constante de temps RC, formée par un condensateur externe et une résistance interne. Malheureusement, le RESET du 8051 est actif à l'état haut (ce qui est un cas rare). Le RC doit donc être câblé en déivateur, comme dans les monostables. On en déduit qu'il ne fonctionne bien que si le temps de montée de l'alimentation est faible devant la constante de temps, et ce n'est pas nécessairement le cas.

De plus, dans les versions "classiques" du 8051, l'état des sorties est mal défini pendant toute la durée du démarrage de l'oscillateur, plus les 24 premiers cycles.

8051 évolués: entrée-sortie Reset

circuit auxiliaire extérieur



Dans les variantes du 8051 possédant un "watchdog", un transistor interne force la ligne de RESET dans l'état actif.

Ce RESET interne doit être répercute sur tout le système externe, car il est inconcevable qu'un périphérique continue à fonctionner alors que le processeur est réinitialisé. Le RESET est donc dans ce cas une sortie, en plus d'être une entrée.

Pour résoudre le problème de conflit entre deux sorties, on a recours à la technique du "drain ouvert". L'état inactif (ici LO) du RESET est fixé par une résistance "pull-down". L'état actif peut être forcé

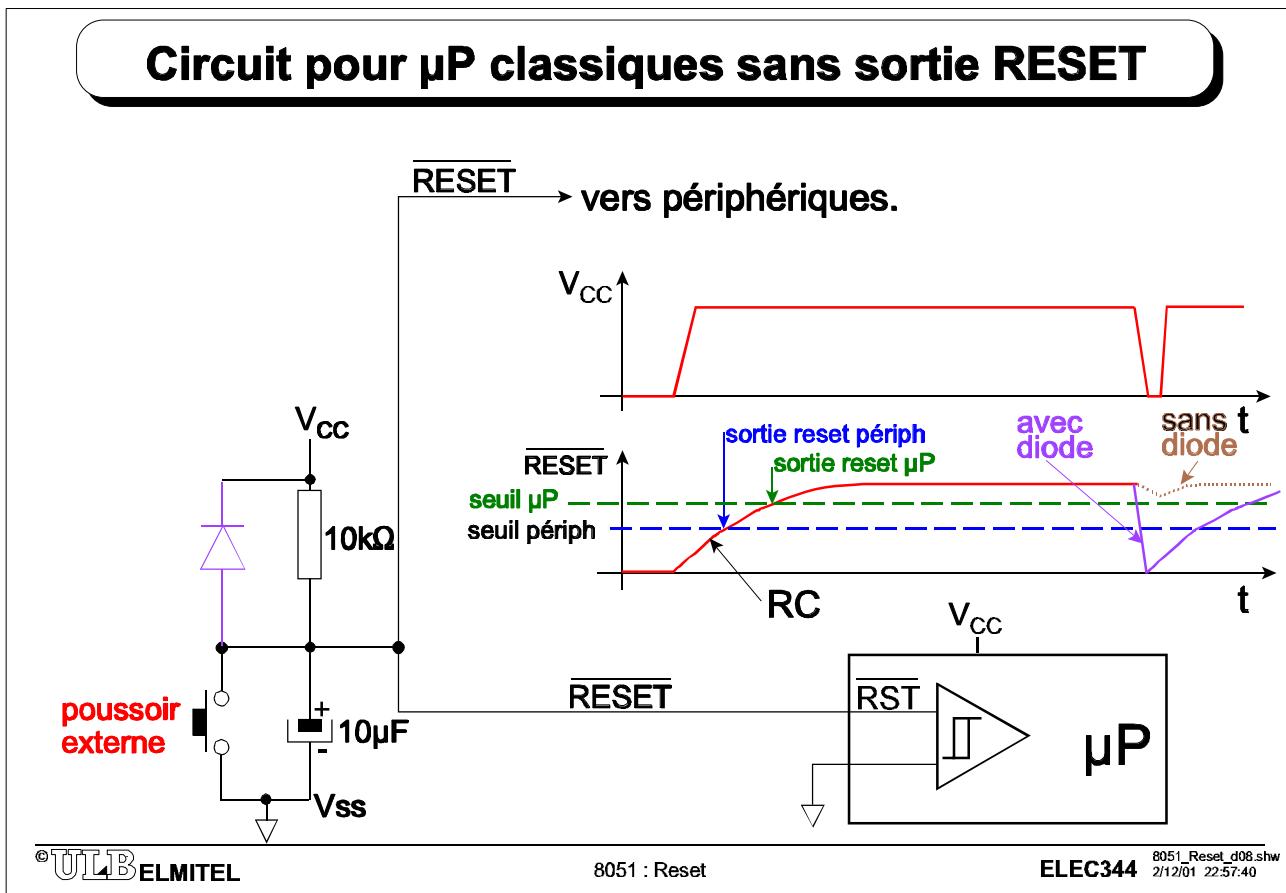
- soit par le circuit externe
- soit par le processeur, via un transistor "pull-up"

Plusieurs constructeurs ont développé des circuits intégrés spéciaux destinés à provoquer le RESET

- à la mise sous tension
- lors d'une chute de tension d'alimentation (la ligne "sense" indiquée sur la figure est une mesure de tension qui est comparée en interne à un seuil précis dans la notice)
- lorsque que l'on agit sur un bouton poussoir ad hoc

L'emploi de tels circuits est tout à fait recommandable. Il en existe plusieurs versions, suivant la tension d'alimentation. Le type 7705 indique ici que le circuit est prévu pour les processeurs alimentés en 5V.

Circuit pour μ P classiques sans sortie RESET

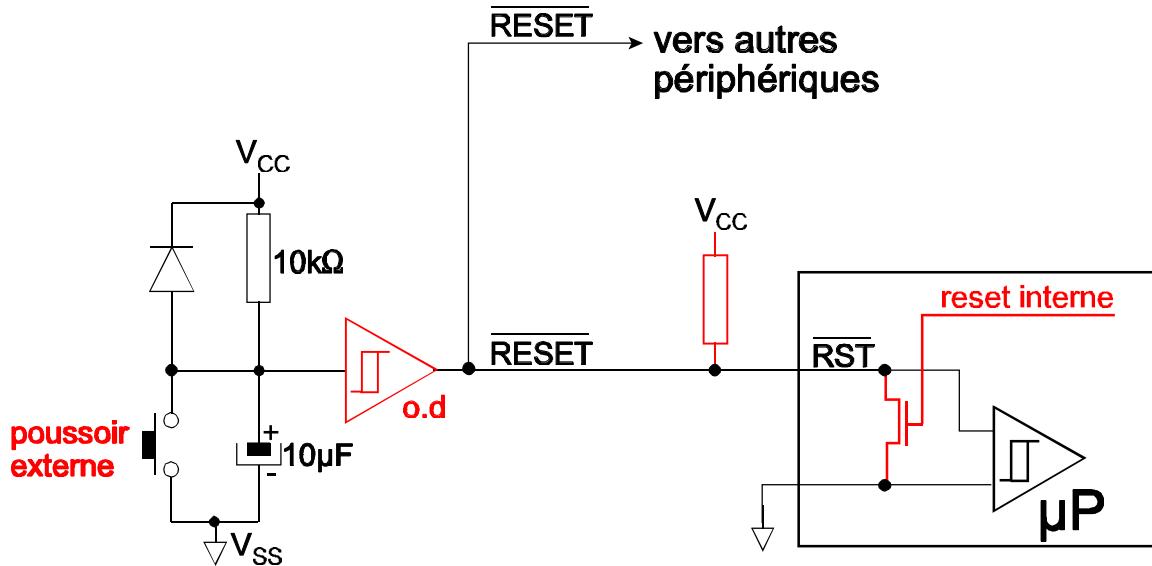


Dans la grande majorité des processeurs autres que le 8051, le RESET est actif à l'état LO. Dans ce cas, le circuit RC fixant le délai est câblé en intégrateur, ce qui simplifie le maintien de l'état actif LO.

Le signal RESET' est distribué dans tout le système. Il convient que tous les périphériques quittent leur RESET avant le processeur. En effet, si ce n'était pas le cas, le processeur pourrait essayer de configurer un périphérique qui ne serait pas encore prêt. Le moyen d'y parvenir est de décaler le seuil logique de sortie du RESET du processeur plus haut que celui des autres circuits du système.

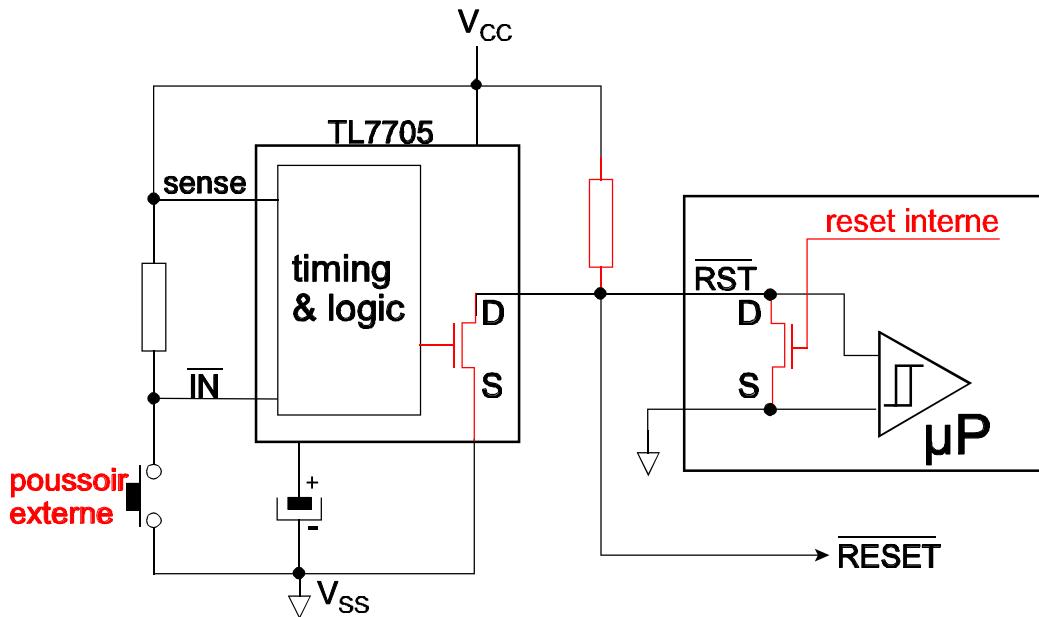
Le poussoir manuel se met directement en parallèle sur le condensateur. En cas de brève coupure de l'alimentation (c'est-à-dire d'une durée inférieure à la constante de temps RC), le processeur sera perturbé (par exemple, il perdra le contenu de ses registres). Il faut alors impérativement que le RESET soit activé, or la constante de temps RC va s'y opposer. L'ajout d'une diode en parallèle sur la résistance assure que la tension du condensateur ne sera jamais supérieure à l'alimentation V_{CC} (au seuil de la diode près). Si V_{CC} chute, le condensateur se déchargera en même temps, activant ainsi la borne RESET.

Circuit pour µP classiques avec sortie RESET



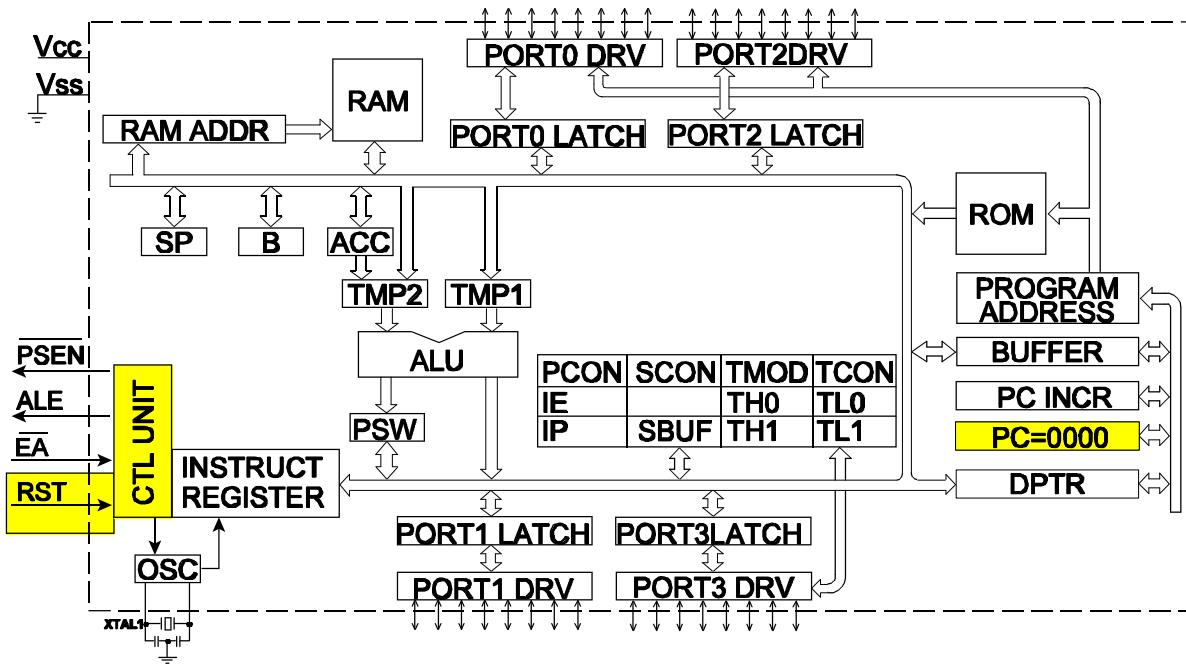
Dans le cas où le processeur possède un RESET d'origine interne (et donc une borne de RESET qui est à la fois une entrée et une sortie), il suffit d'interposer une porte à drain ouvert et une résistance de "pull-up" entre le RC qui fixe le délai et le processeur.

Circuit auxiliaire de Reset



Les circuits intégrés spéciaux sont adaptés au RESET actif HI, actif LO, voire les deux s'il y a deux sorties à drain ouvert.

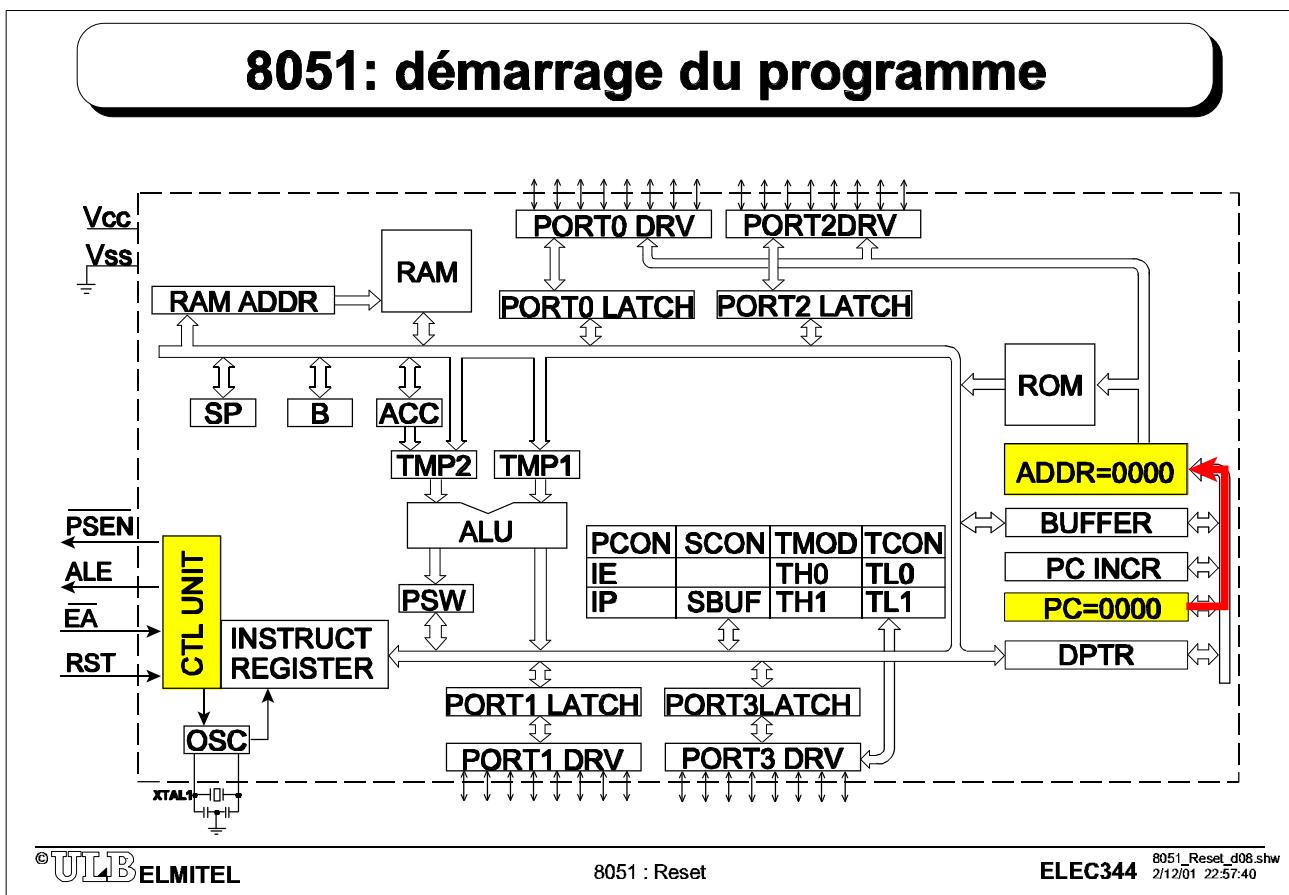
8051: effet du RESET sur le PC



Le compteur de programme PC ("Program Counter") est l'un des registres fondamentaux du processeur. Ce registre est un groupe de D flip-flops de la taille du bus d'adresses (16 bits dans le cas du 8051) et contient l'adresse de la prochaine lecture à faire en mémoire programme.

Le RESET doit donc avoir pour effet de positionner le PC à l'adresse de la première instruction; pour le 8051, c'est l'adresse 0.

8051: démarrage du programme



©ULB ELMITEL

8051 : Reset

ELEC344 8051_Reset_d08.shw
2/12/01 22:57:40

21

Lorsque l'on désactive le RESET, l'unité de contrôle quitte son état de repos et entame la séquence d'opération par le transfert du compteur de programme PC dans le registre qui détermine l'adresse à présenter à la mémoire programme ROM.

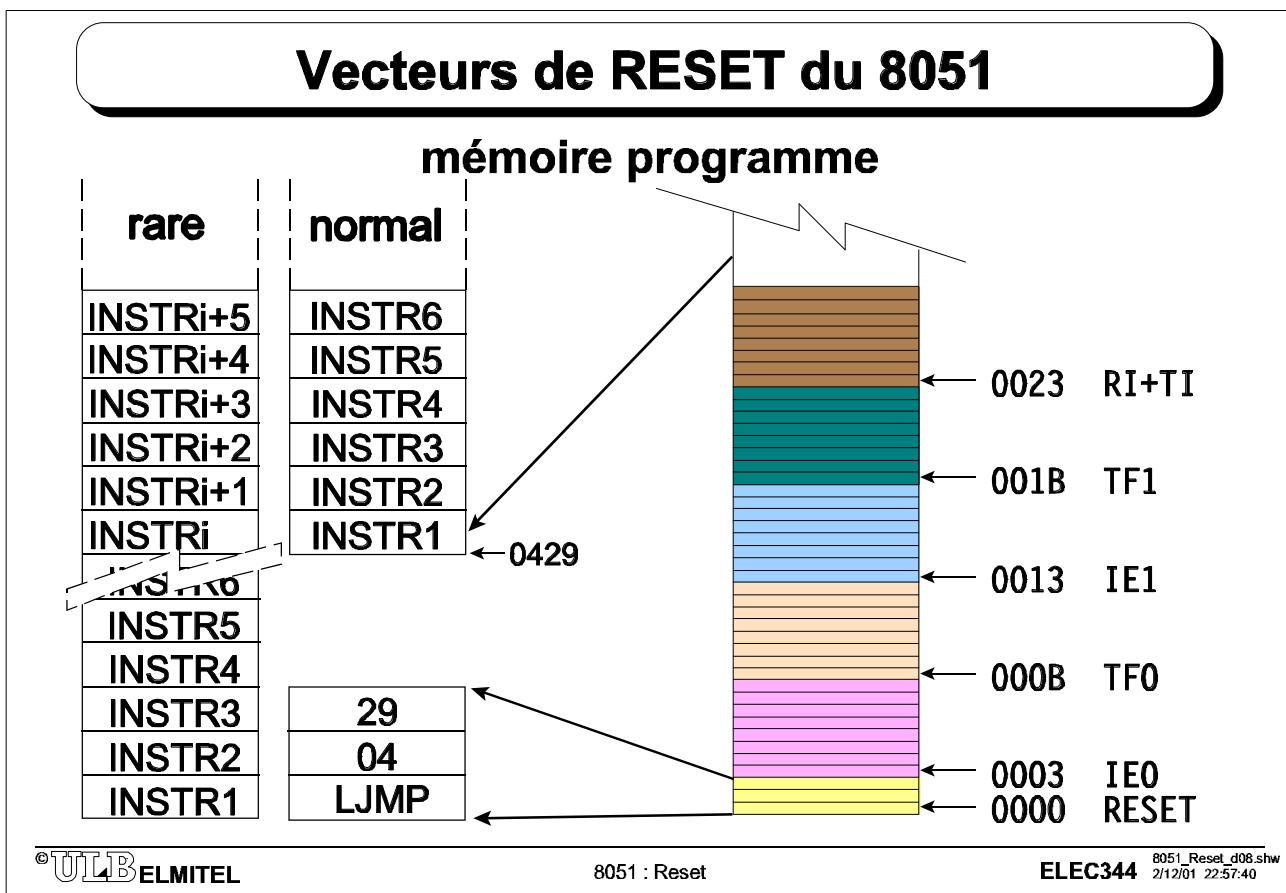
22

Après le Reset .. notion de vecteur de RESET

- ▶ PC = valeur imposée par le RESET (ex. 0000)
- ▶ PC => Adresse ROM
- ▶ lecture en ROM : 2 options
 - ◆ 8051 : lecture du premier code exécutable
 - ◆ µP classique :
 - lecture de l'**adresse** du premier code exécutable ou "vecteur de RESET"
 - saut implicite à cette adresse

- dans le cas du 8051, la première lecture en ROM ramène un mot qui code la première instruction exécutable du programme
- dans la plupart des autres processeurs, les premiers mots extraits de la ROM constituent l'adresse de la première instruction exécutable. Cette adresse porte le nom de "vecteur de RESET". La première instruction à exécuter est donc implicite; c'est un saut à l'adresse de ce vecteur.

Vecteurs de RESET du 8051



25

La figure représente la structure du début de la mémoire programme d'un 8051. En théorie, le programme peut être constitué d'une suite d'instructions commençant à l'adresse 0, mais ce cas est rare.

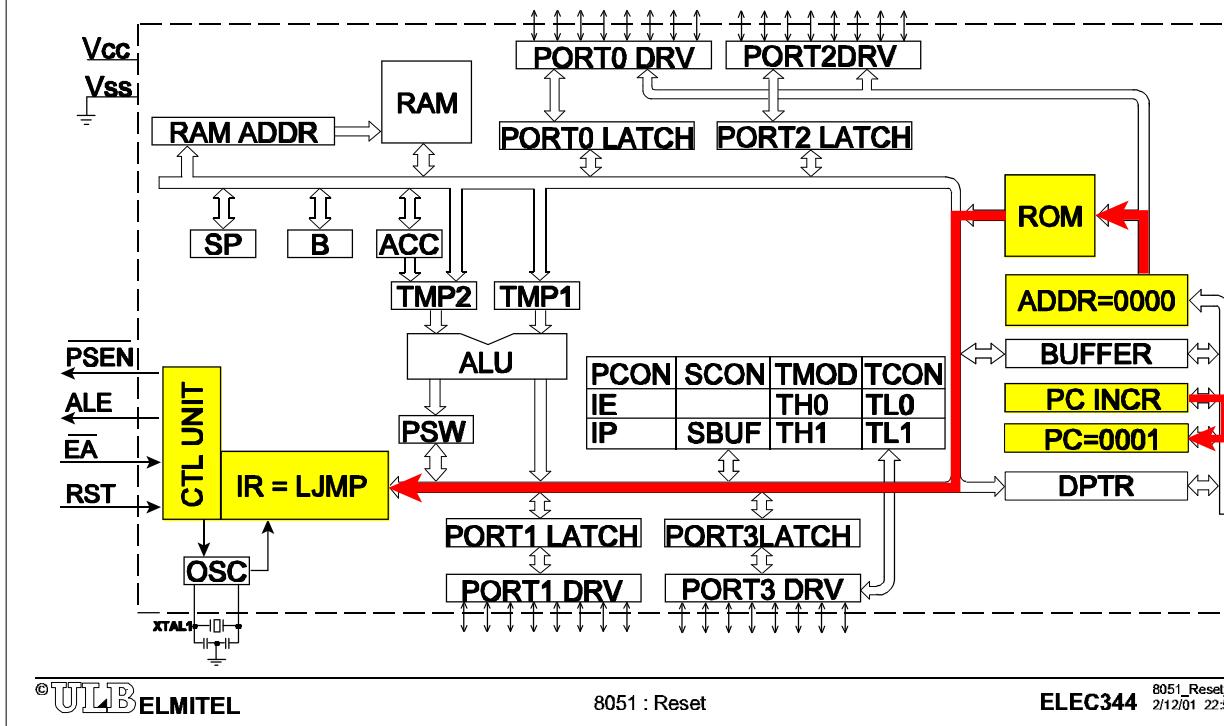
Dans la grande majorité des programmes, la première instruction est un saut, et les deux octets suivants codent l'adresse de destination de ce saut qui, elle, est l'adresse de la première instruction exécutable (dans l'exemple 0429H).

Une des raisons de ce saut est de ménager de la place pour une table d'autres vecteurs dits "vecteurs d'interruption" qui pointent vers le début de la portion code à exécuter lors de l'occurrence d'une interruption interne ou externe. (voir la leçon sur les interruptions).

La suite de ce chapitre sera donc basée sur ce dernier cas puisqu'il est nettement plus fréquent.

26

8051: décodage de la 1ère instruction



Conformément à ce que nous venons de dire, l'opération de lecture extrait de la mémoire programme (ROM) un mot de 8 bits codant l'instruction de saut LONG JUMP (LJMP).

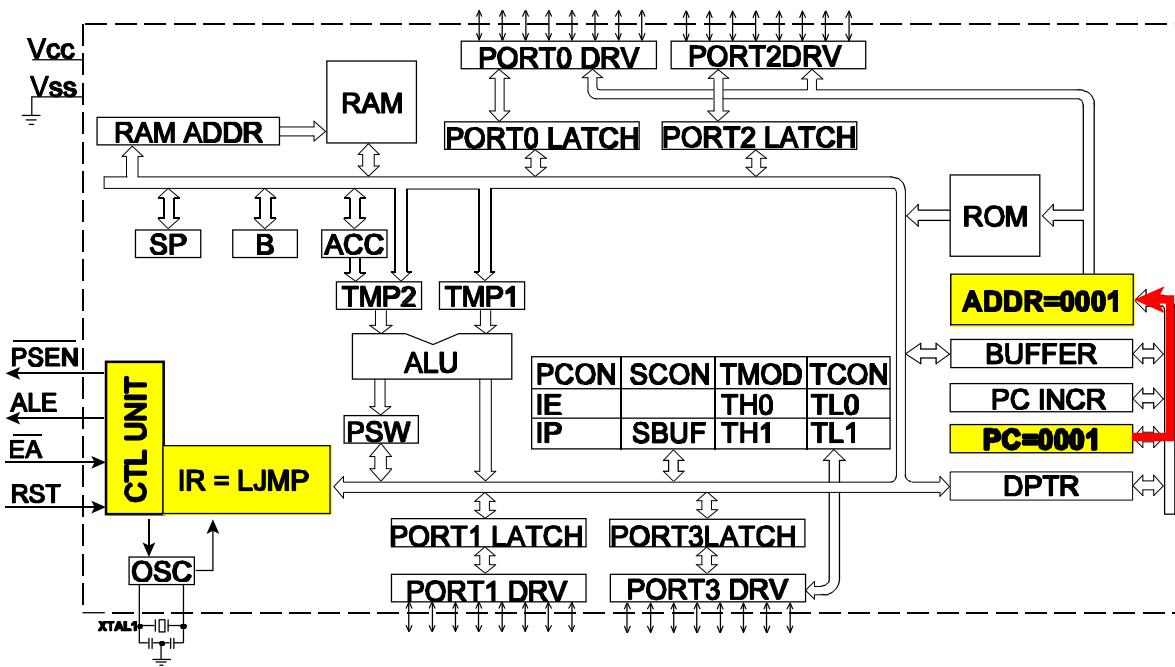
Cette opération porte le nom d'*extraction* ou "op-code fetch".

Le mot provenant de la mémoire programme est transmis par un bus interne vers un registre spécial appelé "registre d'instruction" ou "instruction register (IR)".

L'unité de commande va alors entamer la phase de *décodage* du contenu de IR, pour déterminer à quelle instruction il correspond et quelle est la séquence d'opérations à effectuer pour l'exécuter.

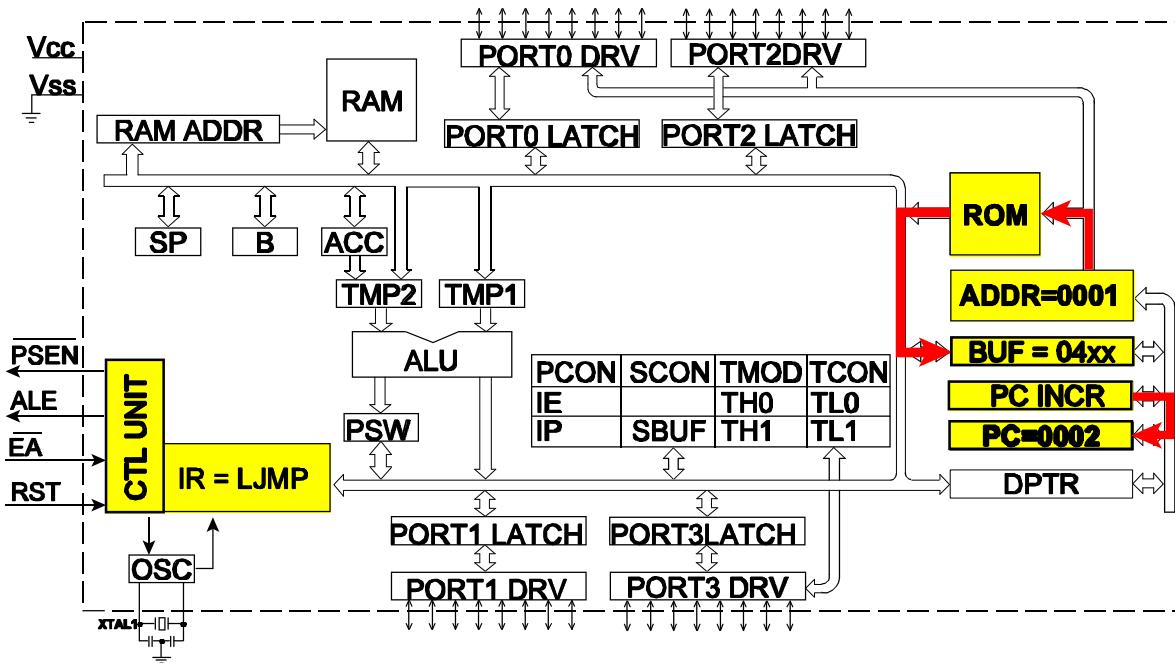
Pendant ce temps, le compteur de programme PC est incrémenté par un additionneur spécial (PC INCR).

8051: préparation de la 2^{ème} lecture en ROM



La nouvelle valeur du compteur de programme PC est ensuite mémorisée dans le registre d'adresse de la mémoire programme

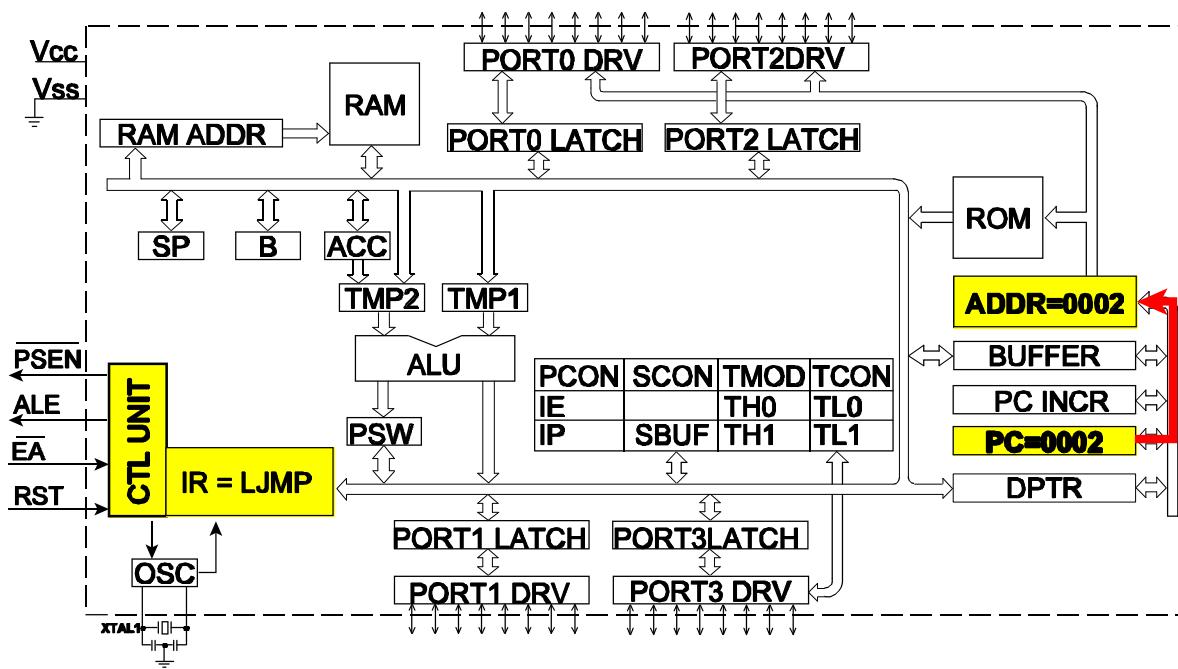
8051 : 2^{ème} lecture en ROM



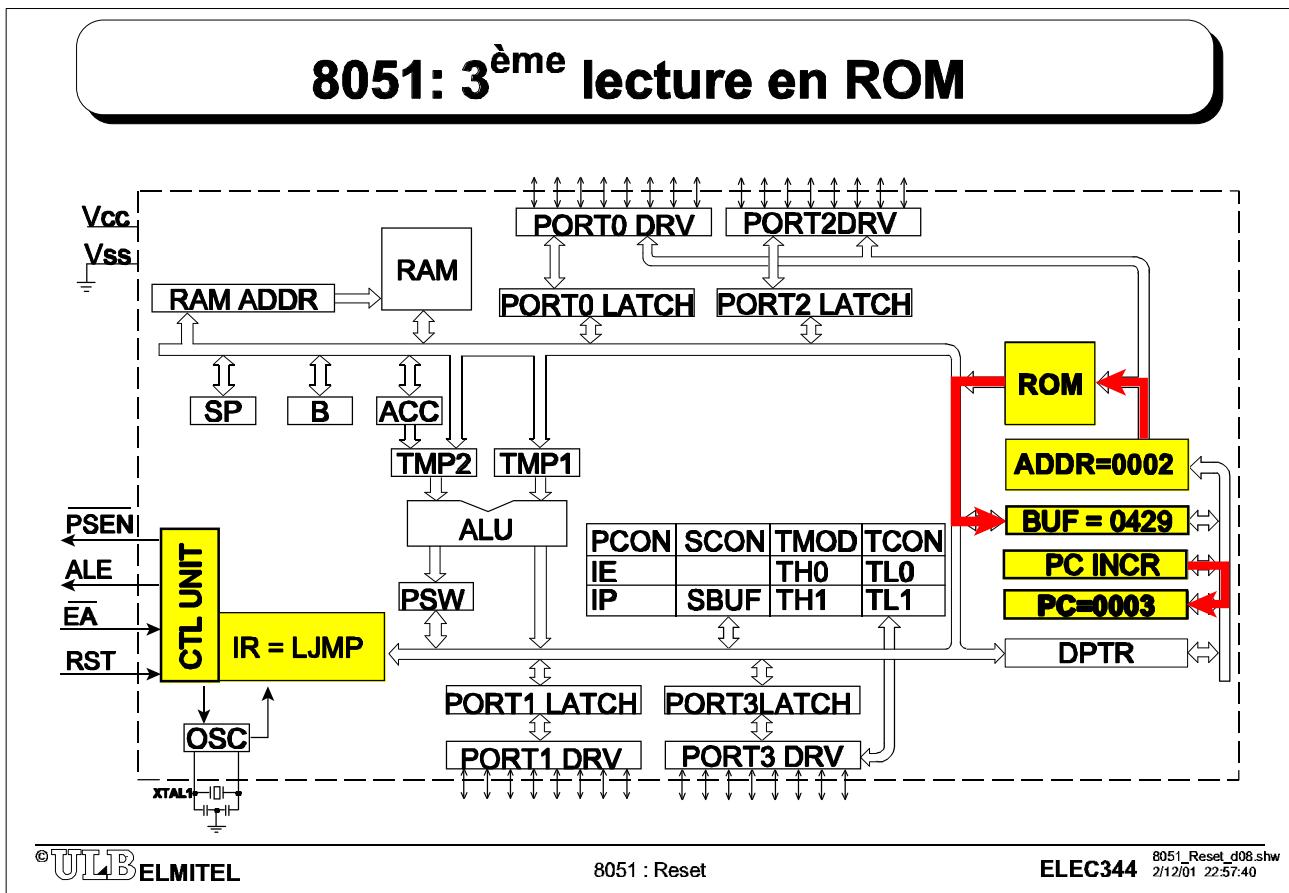
Le décodage de l'instruction LJMP détermine l'étape suivante de la machine d'état de l'unité de commande : provoquer une deuxième lecture dans la mémoire programme, pour en extraire le mot de poids fort du vecteur de RESET (ici 04H). Ce mot n'étant pas une instruction, il n'est plus stocké dans le registre d'instruction IR, mais bien dans un tampon (BUFFER) où il attendra d'être complété.

La troisième lecture en ROM se prépare par l'incrémentation du compteur de programme PC.

8051: préparation de la 3^{ème} lecture en ROM



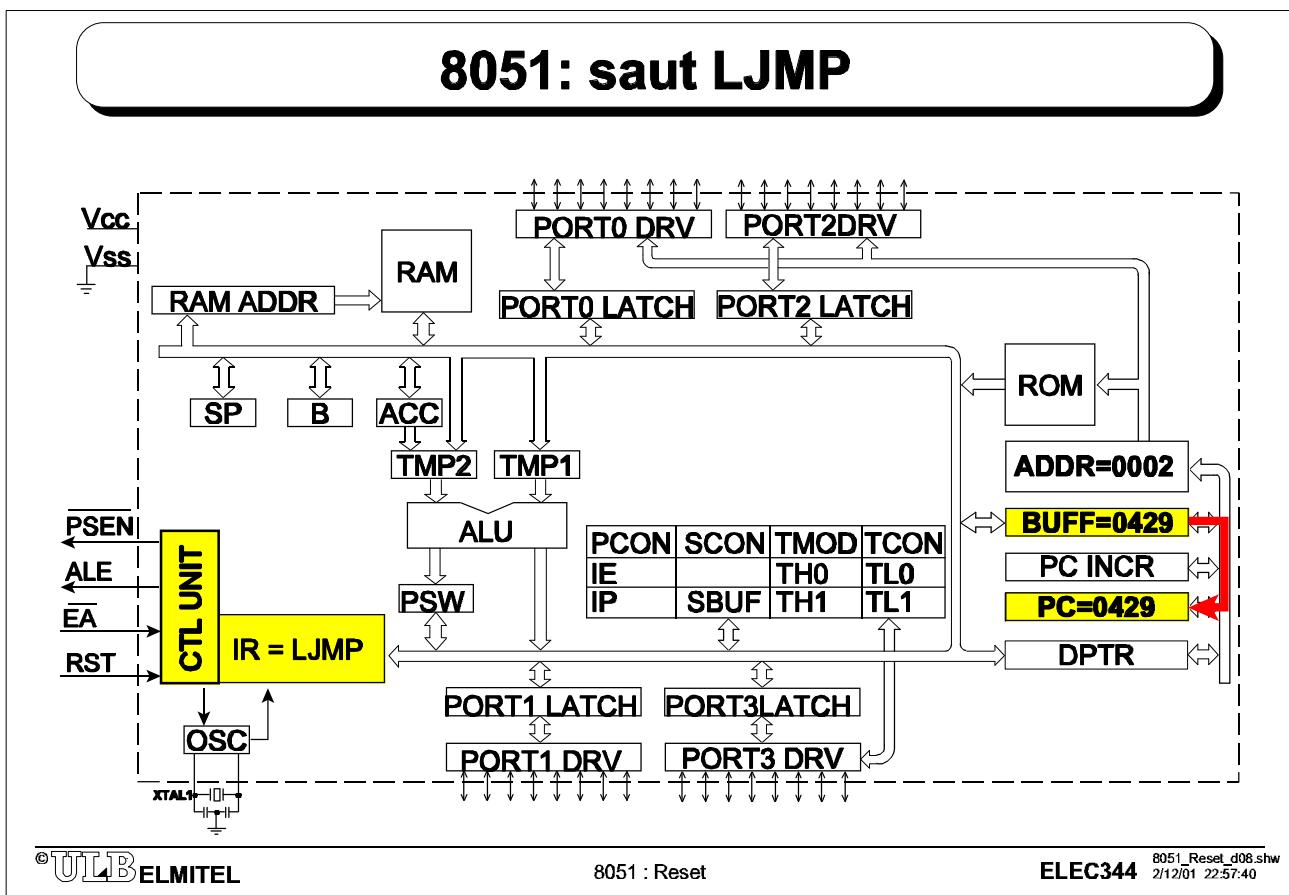
8051: 3^{ème} lecture en ROM



La troisième lecture en ROM ramène la 2^{ème} partie du vecteur de RESET, qui complète le buffer ; l'unité de commande connaît désormais l'adresse de la première instruction utile (0429H dans l'exemple).

Le compteur de programme PC est automatiquement incrémenté.

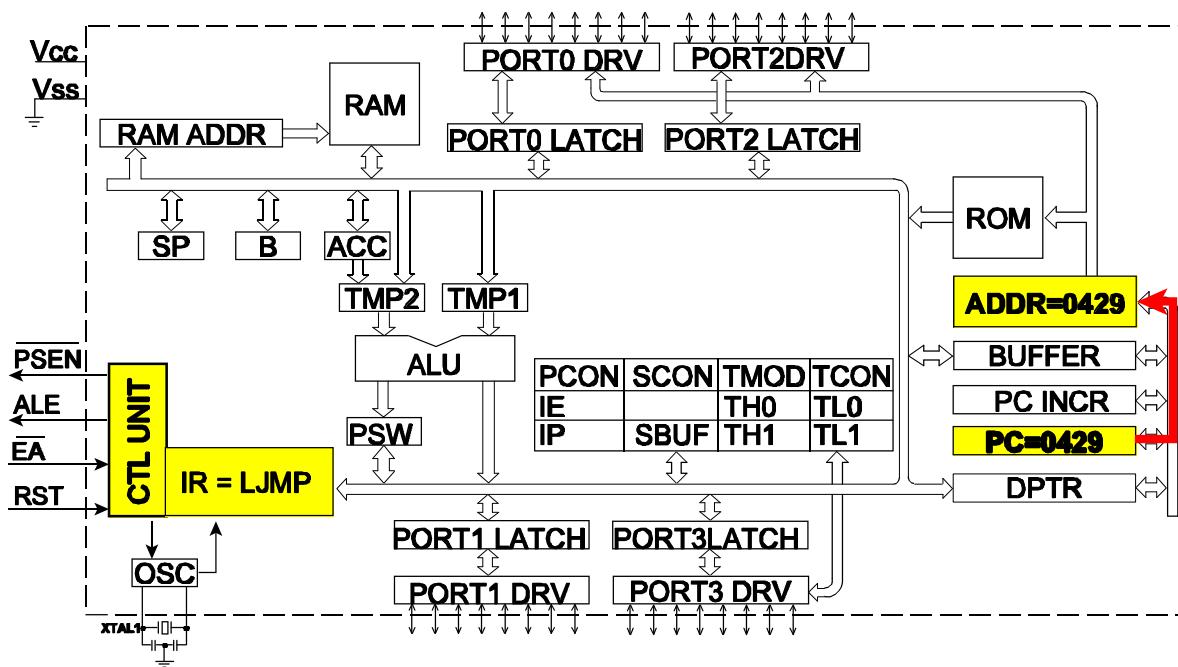
8051: saut LJMP



Normalement l'exécution du programme est linéaire, c'est-à-dire formée d'une suite d'instructions consécutives en mémoire programme, d'où l'incrémentation automatique du compteur de programme PC.

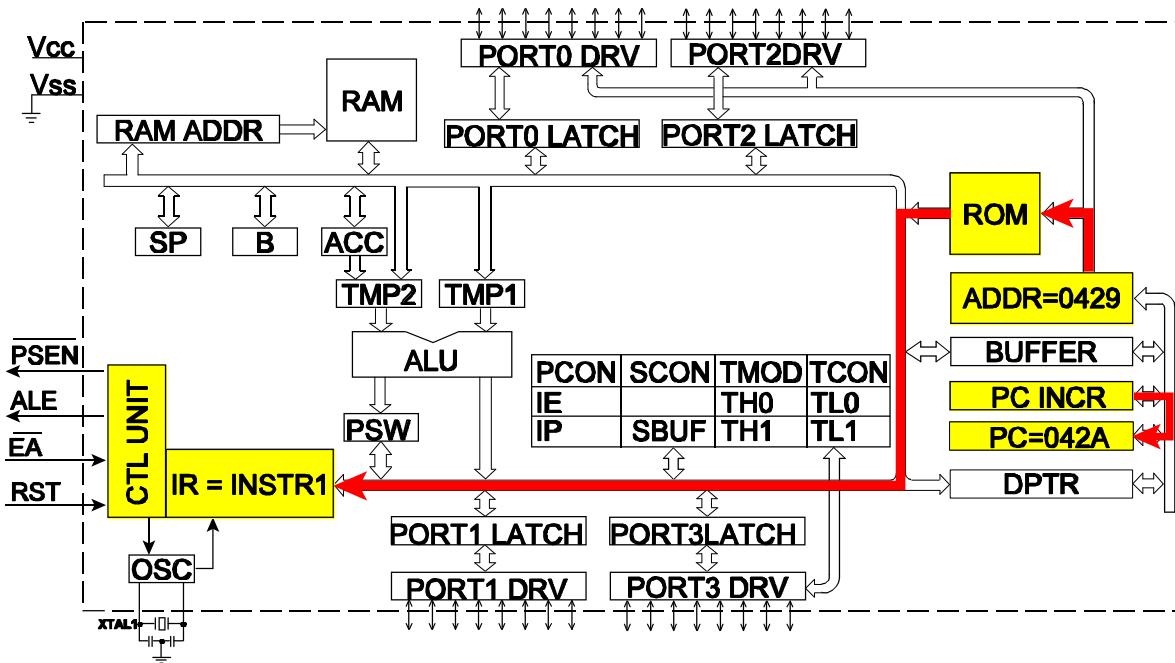
L'instruction LJMP est une rupture de séquence, qui se traduit à l'intérieur du processeur par le transfert du contenu du "buffer", vers le compteur de programme PC, écrasant ainsi sa valeur précédente.

8051: préparation de la 4^{ème} lecture en ROM



Le processus normal reprend avec le nouveau compteur de programme, qui est transféré dans le registre d'adresse.

8051 : première instruction du programme



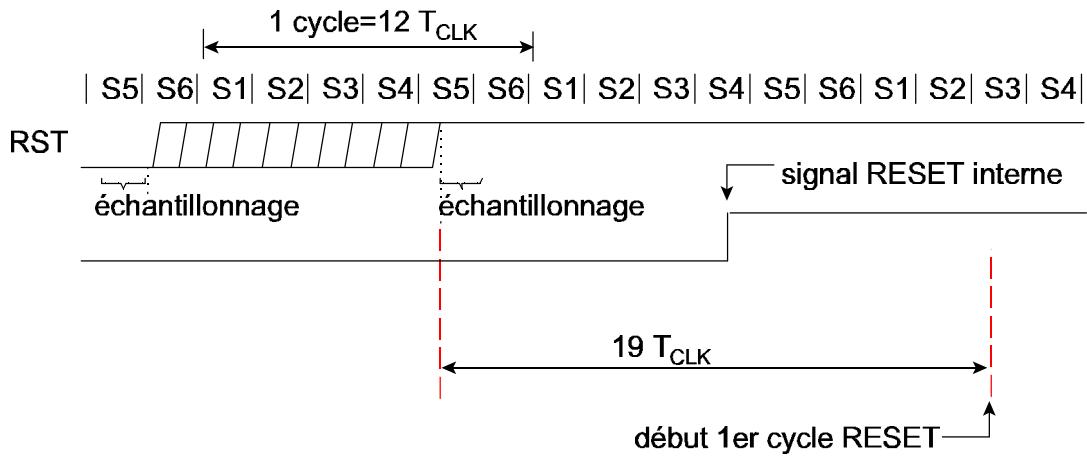
La quatrième lecture en ROM se fait à l'adresse de la première instruction exécutable (ici 0429H) et le code de cette instruction est placé dans le registre d'instruction (IR), pour être décodé à son tour.

Le compteur de programme PC est incrémenté automatiquement.

A partir de là, le programme se poursuit dans une séquence systématique :

- extraction d'une instruction en mémoire programme (OPCODE FETCH)
- décodage de l'instruction, suivi au besoin de lecture(s) supplémentaire(s) en mémoire programme pour fournir des compléments d'instruction (constante, adresse...)
- exécution proprement dite

Chronogramme de RESET



Cette figure montre que le signal externe de RESET est resynchronisé sur le fonctionnement du processeur.

Le cycle machine du 8051 comprend 12 périodes d'horloge, regroupés en 6 stades S1 à S6. Le signal de RST est échantillonné pendant S5.

Sur la figure, le signal RST est activé trop tard pour être pris en compte au premier cycle. Pour qu'il soit effectif au cycle suivant, le signal RST doit être activé au plus tard un "setup time" avant S5. Après cela, le signal interne de RESET est activé au stade S4 du cycle suivant; la première instruction de RESET sera activée 19 périodes d'horloge après la prise en compte du signal externe.