Wikipédia

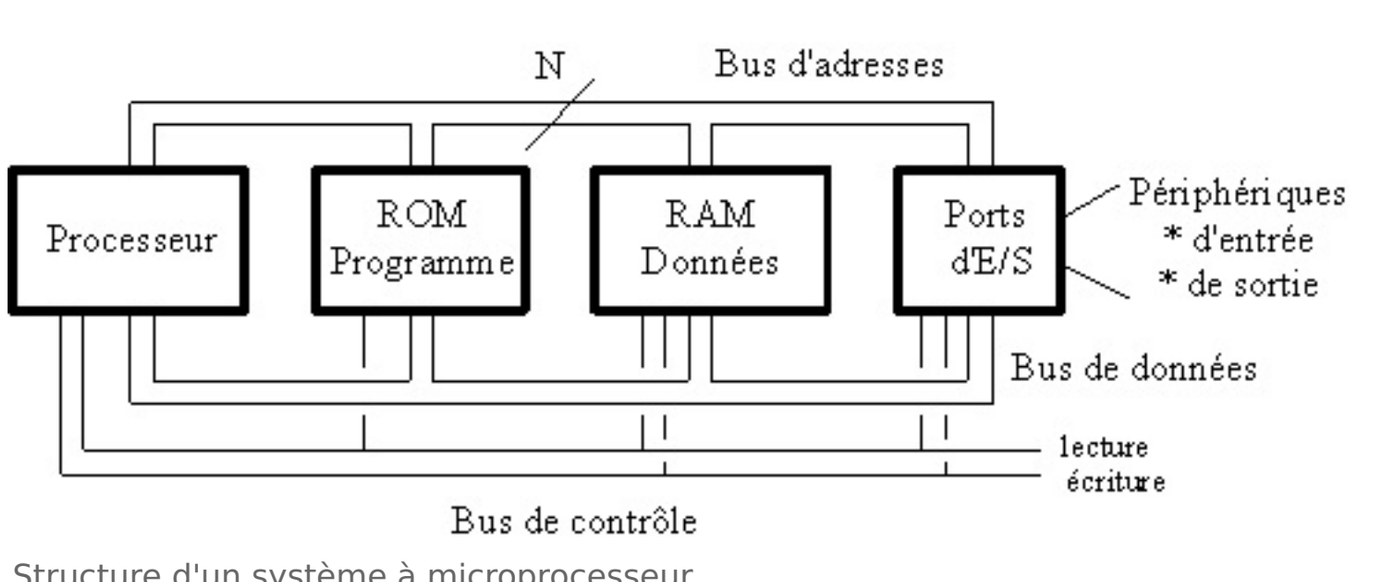
Intel 8051

Intel 8051 ou **8051** est un microcontrôleur (MC) développé par Intel en 1980 pour être utilisé dans des produits embarqués. C'est encore une architecture populaire ; de nombreux microcontrôleurs plus récents incorporent un cœur 8051, complété par un certain nombre de circuits périphériques intégrés sur la même puce, et dotés de mémoires de plus grande capacité.

Intel P8051.

La structure d'un système à microprocesseur (MP) est indiquée dans la figure ci-après.

- Le MP, parfois appelé **unité centrale**, est le cœur du système, puisque c'est lui qui est chargé d'exécuter les instructions du programme.
- Le programme est inscrit dans la mémoire ROM du système ; il s'agit d'une mémoire non volatile, c'est-à-dire qui préserve les informations qu'elle contient même en l'absence d'alimentation électrique.
- La mémoire vive ou RAM sert à stocker des informations durant le fonctionnement du programme : informations en provenance de l'extérieur, données intermédiaires dans les calculs...
- Les ports d'entrées/sorties permettent la communication du système avec le monde extérieur ; les périphériques d'entrée seront par exemple des boutons-poussoirs, des interrupteurs, des commutateurs, des compteurs, des convertisseurs analogiques/numériques... ; les périphériques de sortie seront des témoins à leds, des relais, des moteurs, des convertisseurs numériques/analogiques... En général, un circuit d'interface de puissance est placé entre le port de sortie et le périphérique commandé.



Structure d'un système à microprocesseur

Un microcontrôleur (MC) réunit ces différentes fonctions électroniques sur une même puce. Dans les paragraphes qui suivent, nous allons décrire les principales parties du microcontrôleur 8051. Le 8051 est un MC 8 bits car il traite des données sur 8 bits ; le bus de données comporte donc 8 lignes ; comme la plupart des MC 8 bits, le 8051 gère des adresses en 16 bits, ce qui donne un espace adressable de 2^{16} octets soit 64 kio.

Sommaire

Structure de l'unité centrale

L'unité centrale est le cœur du MC. Elle est chargée d'exécuter les instructions du programme. On peut distinguer 3 parties : traitement des données, gestion des adresses, traitement des instructions.

Traitement des données

Cette partie comporte 3 registres et une **unité arithmétique et logique** (UAL). Les instructions du 8051 permettent d'agir sur 1 ou 2 données de 8 bits à la fois, pas plus. Le fonctionnement est le suivant : les registres de 8 bits TMP1 et TMP2 présentent à l'UAL la ou les données à traiter. Après traitement, le résultat est placé dans l'accumulateur (Acc). Il peut être utilisé pour un nouveau calcul, envoyé dans une mémoire (RAM ou registre de travail) ou vers un port de sortie.

L'UAL comporte les circuits suivants :

- un additionneur pour 2 nombres de 8 bits ; le 9e bit de l'addition est placé dans l'indicateur d'état C (Carry) ;
- un soustracteur pour 2 nombres de 8 bits ; l'emprunt éventuel généré si le résultat est négatif est aussi placé dans un indicateur d'état ;
- un circuit de multiplication pour 2 nombres de 8 bits ; ici, le résultat peut nécessiter jusqu'à 16 bits ; les 8 bits les plus significatifs sont placés dans un registre B prévu à cet effet ;
- un circuit de division pour 2 nombres de 8 bits ; il s'agit ici d'une division entière : le quotient est placé dans l'Acc, le reste dans le registre B ;
- 8 circuits logiques de type inverseur, ET, OU, XOU ;
- de nombreux tampons 3 états pour permettre les communications entre les registres, l'UAL et le bus de données aux moments adéquats.

Une série de bascules, appelées **indicateurs d'état**, sont automatiquement positionnées lors des opérations arithmétiques. L'état de ces indicateurs peut être testé par les instructions de branchement conditionnel.

Le 8051 comporte 32 registres de travail, organisés en 4 groupes de 8 registres. Ces registres constituent des mémoires temporaires, où l'on place provisoirement des informations dont on sait qu'elles seront nécessaires peu de temps après. L'intérêt des registres de travail, c'est que le temps d'accès est plus court que pour une lecture ou écriture en RAM, et les instructions nécessaires pour y accéder plus courtes.

Gestion des adresses

L'espace mémoire d'un 8051 est partagé en 3 zones bien distinctes :

- la **zone programme** ; pour assurer sa permanence en l'absence d'alimentation, le programme est mis en mémoire ROM ; on peut également placer dans la ROM, en même temps que le programme, des données permanentes qui pourraient être utiles : facteurs de conversion pour des calculs, informations diverses tels que noms de jours de la semaine, des mois...
- la **zone données**, en RAM ;
- la **pile** : il s'agit d'une zone de RAM qui est utilisée principalement pour préserver les adresses de retour des sous-programmes ; on peut également l'utiliser comme zone de stockage temporaire (un peu comme les registres de travail) ou pour l'échange de données entre programme principal et sous-programmes.

Le 8051 comporte donc très naturellement trois registres consacrés à la gestion des adresses.

- Le **pointeur de programme**, aussi appelé compteur ordinal (PC, *Program Counter*), est un registre de 16 bits qui contient à tout moment l'adresse de la prochaine instruction à exécuter. Il est mis à 0 au moment de la mise sous tension du système ; le programme doit donc impérativement commencer à l'adresse 0000h (h signifiant code hexadécimal, c'est le code le plus couramment employé pour définir les zones d'adresse dans les MP et MC). La plupart du temps, ce pointeur est simplement incrémenté chaque fois que l'on va chercher dans la ROM un octet du programme (une instruction comporte de 1 à 3 octets). Toutefois, le contenu du pointeur est complètement modifié lorsque l'on doit effectuer un branchement (saut ou sous-programme), puisque c'est alors l'adresse de l'endroit où l'on veut aller qui y est introduite. Pour permettre au MP de revenir à l'endroit de départ dans le cas des sous-programmes, on sauvegarde dans la pile l'adresse à laquelle on était avant d'effectuer le branchement.
- Pour lire ou écrire une donnée dans la RAM, l'adresse souhaitée doit se trouver dans le **pointeur de données** (DPTR, *Data Pointer*).
- Enfin, le programmeur définit la zone de RAM qu'il va attribuer à la pile en introduisant, au démarrage du programme, l'adresse de départ de la pile dans le **pointeur de pile** (SP, *Stack Pointer*). Le pointeur indique à chaque instant la plus haute adresse occupée dans la pile. Il est incrémenté au moment où l'on veut charger un nouvel octet dans la pile, et décrémenté si l'on retire un octet. Par défaut, le pointeur est initialisé à 07h au démarrage. Le pointeur peut aussi être modifié par le programme.

Traitement des instructions

Les instructions comportent de 1 à 3 octets. Le premier octet, appelé **code opératoire**, précise de quelle instruction il s'agit. Certaines instructions nécessitent une donnée (8 bits) ou une adresse restreinte (8 bits), voire une adresse absolue (16 bits).

Le code opératoire est placé dans le registre d'instructions, prévu à cet effet. Il est présenté à un décodeur qui détermine le type d'instruction et, le cas échéant, ira lire 1 ou 2 octets supplémentaires qui seront placés dans des registres *ad hoc* (donnée ou adresse).

Le bloc traitement des instructions est piloté par une horloge ; le 8051 contient tous les éléments de l'horloge, à l'exception des composants qui déterminent la fréquence d'oscillation : généralement un quartz, mais un circuit RC peut aussi être utilisé si une connaissance précise de la fréquence n'est pas nécessaire. Le bloc génère des signaux de contrôle (lecture, écriture) tant pour les autres blocs du MC que pour les circuits extérieurs éventuels (mémoires externes...).

Mémoires

- Le 8051 comporte une mémoire ROM de 4Ko ; cette mémoire doit être programmée par le constructeur, au moment de la fabrication du composant. D'autres membres de la famille du 8051, tels le 8751 remplacent la ROM par une EPROM, qui peut être programmée (et effacée) par l'utilisateur. Le coût est plus élevé, mais ce composant est fort utile dans la phase de mise au point du programme puisqu'il permet de modifier le programme.
- Le 8051 comporte une mémoire RAM de 128 octets. Celle-ci comporte 3 zones :
  - les 32 octets de 00h à 1Fh peuvent être utilisés comme 4 groupes de 8 registres de travail ;
  - les 16 octets de 20h et 2Fh sont utilisés pour les variables booléennes, c'est-à-dire des variables d'un bit ;
  - les octets 80 qui restent, de 30h à 7Fh, constituent la RAM proprement dite ; rappelons qu'il faudra réserver une zone de RAM pour la pile, en général quelques octets dans le haut de la mémoire.

Remarque : les octets qui ne sont pas utilisés comme registres de travail ou pour les variables booléennes peuvent être utilisés comme RAM conventionnelle.

- Le 8051 comporte aussi un certain nombre de registres appelés SFR (*Special Function Registers*). Ces registres contrôlent le fonctionnement des divers périphériques intégrés : ports d'entrées/sorties, compteurs/temporisateurs etc.

La tendance, dans les systèmes utilisant un MC, est de choisir un composant disposant d'une ROM et d'une RAM de capacité suffisantes pour stocker l'entièreté du programme et l'ensemble des données. Toutefois, si nécessaire, on peut adjoindre au MC des mémoires externes. Le MC se comporte alors comme un microprocesseur. Des lignes qui sont normalement utilisées comme ports d'entrées/sorties sont converties en bus d'adresses et de données. Dans ce type d'applications, on choisira généralement des MC sans ROM interne, tels le 8031.

Modes d'adressage

Le 8051 admet 6 modes d'adressage :

- l'adressage implicite ;
- l'adressage registre, qui est utilisé pour spécifier un des 8 registres de travail du groupe actif (le groupe actif est spécifié par 2 bits dans un des SFR ; à la mise sous tension, c'est le groupe 0 qui est spécifié ;
- l'adressage direct restreint : utilisé pour accéder à la RAM et aux SFR ;
- l'adressage indirect à registre : utilisé pour accéder à la RAM interne, externe et aux ports d'E/S ; pour les adresses en 8 bits, c'est le registre de travail R0 ou R1 qui joue le rôle de pointeur de données ; pour les adresses en 16 bits, c'est le DPTR (*Data Pointer*) ;
- l'adressage immédiat, où la donnée suit le code opératoire ;
- l'adressage indexé : c'est une variante de l'adressage indirect à registre : l'adresse de la donnée est calculée en additionnant les contenus de l'Acc et du pointeur de programme (Acc + PC) ou du pointeur de données (Acc + DPTR).

Jeu d'instructions

Dans cette section, nous donnons un aperçu du jeu d'instructions du 8051 en employant la syntaxe du constructeur, Intel :

- Rr : registre de travail R0 à R7 du groupe sélectionné
- direct : adresse directe (RAM ou SFR)
- @Ri : case de RAM pointée par R0 ou R1
- @DPTR : case mémoire pointée par le DPTR
- #data : donnée immédiate 8 bits
- #data16 : donnée immédiate 16 bits
- bit : adresse bit (dans les 16 octets pour variables booléennes et les SFR)
- addr16 : adresse de destination pour les branchements
- rel : adresse relative pour les branchements ; l'adresse est indiquée en complément à 2, de façon à pouvoir effectuer des sauts en avant ou en arrière (les nombres de 128 à 255 sont considérés comme des nombres négatifs)

Instructions de transfert

Les adresses de source et de destination peuvent être l'Acc, un registre de travail, une case de RAM interne ou externe, un port d'E/S ; la ROM ne peut servir que de source.

- Transferts en RAM interne

La structure des instructions est : **MOV destination, source** copie l'octet de l'adresse source à l'adresse de destination

  - MOV A,R7 copie le contenu de R7 dans l'Acc
  - MOV 70h,A copie le contenu de l'Acc dans la case de RAM interne à l'adresse 70h
  - MOV A, @R1 copie le contenu de la case dont l'adresse est en R1 dans l'Acc
  - MOV 20h, 71h copie le contenu de la case 71h de RAM interne dans la case 20h

- Transferts de et vers la RAM externe

La structure des instructions est : **MOVX destination, source**

  - MOVX A, @DPTR copie le contenu de la case dont l'adresse se trouve dans le DPTR dans l'Acc
  - MOVX @DPTR, A copie le contenu de l'Acc dans la case dont l'adresse se trouve dans le DPTR

Instructions arithmétiques

D'une façon générale, elles utilisent l'Acc pour stocker une des deux données de départ, ainsi que le résultat de l'opération après exécution de l'instruction. Des indicateurs d'état sont positionnés automatiquement en fonction du résultat de l'opération :

- l'indicateur de dépassement (*Carry*) est mis à 1 si l'addition provoque un report au 9e bit ;
- l'indicateur de semi-dépassement (*Half-Carry*) est mis à 1 si une addition provoque un report au 5e bit (cet indicateur est utilisé par l'instruction DA A, qui permet de faire l'addition de deux nombres en code DCB, décimal codé binaire) ;
- l'indicateur de parité est mis à 1 si la parité du nombre dans l'Acc est impaire ;
- l'indicateur d'*overflow* est positionné lors de certaines opérations arithmétiques.

Exemples

  - ADD A, R5 : ajoute le contenu de R5 à l'Acc
  - ADD A,#17h : ajoute 17h au contenu de l'Acc
  - SUBB A,R2 : soustrait à l'Acc le contenu de R2
  - MUL AB : effectue la multiplication des nombres placés dans l'Acc et dans le registre B ; les 8 bits les moins significatifs sont placés dans l'Acc, les plus significatifs dans B
  - DIV AB divise le contenu de l'Acc par le contenu de B, place le quotient en A et le reste en B
  - INC R0 incrémente le contenu de R0 sans modifier l'Acc

Instructions logiques

Les opérations logiques sont effectuées entre bits de même poids, il n'y a pas d'interaction entre bits de poids différent et les indicateurs d'état ne sont pas modifiés. Les opérations ET, OU et XOU sont disponibles, ainsi que l'inversion ; pour NON-ET, NON-OU, NON-XOU, il faut agir en deux étapes : d'abord l'opération directe, puis inverser tous les bits.

Exemples

  - CPL A : complémente tous les bits de l'Acc
  - ANL A, 35h : réalise une fonction ET entre les bits de l'Acc et ceux de la case 35h

CLR A : met l'Acc à 0

Instructions booléennes

Il s'agit ici d'une innovation par rapport aux microprocesseurs, qui traitent toujours les données par octets. Le 8051 contient un processeur complet agissant sur des données d'1 bit, aussi appelées variables booléennes. C'est le *Carry* qui joue le rôle d'Acc pour ces opérations.

La RAM interne du 8051 contient 128 cases pouvant être adressées individuellement. Un certain nombre de bits des SFR sont aussi adressables individuellement, en particulier les bits correspondant aux ports de sortie.

  - SETB 38h : positionne à 1 le bit 38h
  - CLR C : met à 0 le *Carry*
  - ORL C,20h : réalise un OU logique entre C et le bit 20h
  - MOV C,45h : copie le contenu du bit 45h dans C.

Instructions de branchement

Rappelons que les instructions de branchement permettent de rompre la séquence normale d'exécution d'un programme. On distingue :

- les sauts : on saute d'un endroit du programme à un autre, sans espoir de retour ;
- les sous-routines ou sous-programmes : on part exécuter un sous-programme, puis on revient à l'endroit d'où l'on était parti (l'adresse de retour est sauvegardée dans la pile).

  - LJMP addr16 : provoque un saut à l'adresse indiquée
  - LCALL addr16 : provoque l'exécution du sous-programme qui commence à addr16
  - JZ F8h : recule dans le programme de 8 pas si le contenu de l'Acc est nul (*Jump if Zero*)
  - CJNE A,#20h,F8h : il s'agit ici d'une opération double : on compare les contenus de l'Acc et de la valeur 20h ; on recule de 8 pas dans le programme s'ils ne sont pas égaux (*Compare and Jump if Not Equal*).

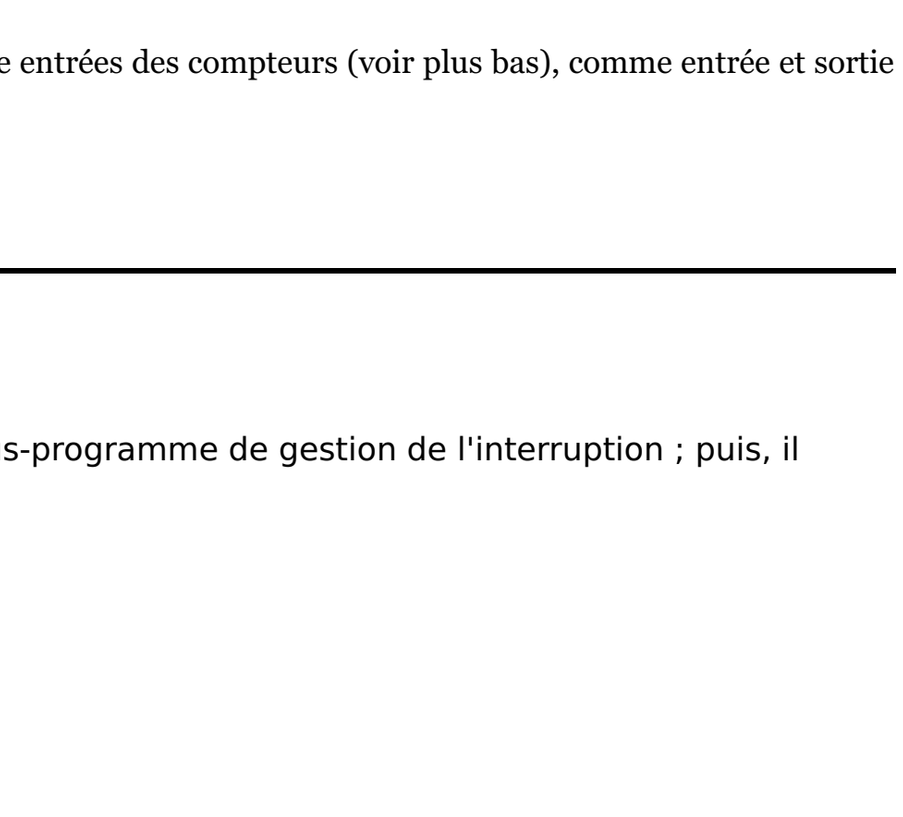
Remarque : seuls les sauts peuvent être conditionnels dans le 8051.

Ports d'E/S

Les ports d'E/S permettent aux systèmes à microprocesseur ou microcontrôleur de communiquer avec le monde extérieur : recevoir des informations, qu'il va ensuite traiter et piloter les périphériques : témoins lumineux, moteurs, relais, convertisseurs N/A etc.

La famille 8051 a été conçue pour des applications nécessitant peu de mémoire ROM et RAM. Ainsi, le 8051 lui-même comporte une ROM de 4Ko et seulement 128 octets de RAM ; et encore faut-il tenir compte que la RAM est utilisée par les registres de travail, la pile et les 16 octets (de 20h à 2Fh) adressables par bit.

Lorsque l'application peut se satisfaire de ces tailles mémoire, 32 lignes d'E/S sont disponibles, organisées en 4 ports de 8 lignes ; les adresses sont les suivantes : port 0 : 80h ; port 1 : 90h ; port 2 ; A0h ; port 3 : B0h. Les circuits internes associés aux différentes lignes sont légèrement différents. Prenons l'exemple du port 1.



Structure d'une ligne du port 1 d'un MC 8051.

Pour utiliser une ligne en ligne de sortie, on écrit le bit désiré dans la bascule D. La sortie Q/ (inverse de Q) pilote la grille d'un mosfet à enrichissement. Si Q est à 0, Q/ est à 1, le mosfet conduit et la ligne de sortie est tirée à 0. Si Q est à 1, Q/ est à 0, le mosfet est bloqué et la ligne de sortie est tirée vers Vcc par la résistance de rappel (*pull-up*).

Pour utiliser la ligne en ligne d'entrée, il faut écrire un 1 dans la bascule D (c'est d'ailleurs l'état par défaut à la mise sous tension). Le circuit extérieur peut tirer la ligne à 0 ; l'état de la ligne apparaît sur le bus de données interne lorsqu'on fait une lecture du port.

Lorsque le système nécessite une plus grande taille mémoire, deux solutions :

- utiliser un membre de la famille doté d'une plus grande capacité mémoire, ou carrément un autre microcontrôleur ;
- adjoindre au 8051 des boîtiers-mémoire externes ; le système accepte des mémoires ROM jusqu'à 64Ko, et des mémoires RAM jusqu'à 64Ko.

Lorsque l'on utilise des boîtiers de mémoire externe :

- on peut choisir un membre de la famille sans ROM interne, comme le 8031 (version du 8051 sans ROM) ;
- le nombre de ports d'E/S disponibles est réduit à 2 (si l'on place uniquement de la ROM en externe), voire 1 (si on place des boîtiers externes de ROM et de RAM ; en effet, les lignes de ports sont utilisées dans cette configuration comme bus d'adresses et de données multiplexés ;
- il faudra donc en général placer des boîtiers de ports d'E/S externes, et leur réserver des adresses pour pouvoir communiquer avec eux.

Remarquons que l'on peut gérer les ports :

- comme des registres 8 bits, en écrivant ou lisant l'état de toutes les lignes d'un port simultanément par les instructions de transfert ;
- comme un ensemble de bascules gérant chacune 1 bit, grâce aux instructions booléennes.

Signalons encore que certaines lignes du port 3 ont une double fonction puisqu'elles peuvent être utilisées comme lignes de demande d'interruption (voir plus bas), comme entrées des compteurs (voir plus bas), comme entrée et sortie du port série (voir plus bas) et comme lignes Read et Write du bus de commande lorsqu'on fait appel à de la RAM externe.

Interruptions

Lorsqu'un périphérique (clavier, convertisseur A/N, port série...) souhaite communiquer une information au MC, il peut :

- attendre que le MC vienne l'interroger ; on parle alors d'**E/S programmées** ;
- demander au MC que celui-ci s'occupe de lui ; on parle alors de **demande d'interruption** ; le MC interrompt la tâche en cours et exécute le sous-programme de gestion de l'interruption ; puis, il reprend la tâche interrompue.

Lorsque le périphérique en question est extérieur au MC, il faut prévoir des broches pour connecter la ou les lignes de demandes d'interruption.

Le 8051 dispose de 5 sources d'interruption :

- deux broches d'interruption externes Int0 et Int1 (broches 2 et 3 du port 3) ;
- deux compteurs/temporisateurs (C/T, voir plus bas) ;
- le port série (voir plus bas).

Chaque interruption peut être masquée ou démasquée en positionnant à 1 ou 0 le bit approprié du registre IE (*Interrupt Enable*). Une interruption masquée n'est pas prise en compte. On peut aussi masquer l'ensemble des interruptions par le bit 7 de IE.

On peut attribuer à chaque type d'interruption une priorité basse ou haute en positionnant le bit approprié du registre IP (*Interrupt Priority*). Une interruption de priorité haute peut interrompre le sous-programme de gestion d'une interruption de priorité basse. Lorsque le MC reçoit simultanément (càd. pendant le même cycle machine) deux demandes d'interruption de même niveau de priorité, celles-ci sont traitées dans l'ordre suivant : Int0, C/T0, Int1, C/T1, port série.

Lorsqu'une demande d'interruption est acceptée, le MC effectue un branchement vers une des adresses suivantes : 0003h pour Int0, 000Bh pour C/T1, 0013h pour Int1, 001Bh pour C/T1 et 0023h pour le port série. Généralement, on place à ces adresses une instruction d'appel de sous-programme pour gérer l'interruption ; à la fin du sous-programme, on revient à l'adresse de départ, qui a été automatiquement mémorisée dans la pile.

Chaque sous-programme d'interruption comporte généralement les parties suivantes :

- sauvegarde des registres de travail qui seront utilisés par le sous-programme ;
- gestion de l'interruption ;
- rechargement des registres sauvegardés ;
- retour au programme principal par l'instruction RETI.

Comme le 8051 comporte 4 blocs de 8 registres de travail, une méthode rapide et élégante pour sauvegarder les registres de travail consiste à simplement changer de bloc actif (celui-ci est déterminé par les bits 3 et 4 du registre d'état, SR, *Status Register*)

À la mise sous tension, les registres IE et IP sont initialisés à 0. Le programmeur doit donc préciser quelles sources d'interruption il veut autoriser, et leur niveau de priorité. Les choix peuvent être modifiés en cours de programme.

Compteurs/Temporisateurs

Le 8051 comporte deux compteurs/temporisateurs. Ces périphériques sont appelés ainsi car ils peuvent être utilisés :

- pour compter les impulsions appliquées à une broche du MC ; on parle alors de **compteur** (*counter*) ;
- pour compter des impulsions provenant de l'horloge du MC ; on parle alors de **temporisateur** (*timer*) puisque dans ce mode le MC peut mesurer des intervalles de temps ou générer des délais précis.

Chaque C/T comporte deux compteurs binaires de 8 bits. Différents modes de fonctionnement sont possibles : les plus utilisés sont :

- le mode 1 : les compteurs sont mis en cascade pour former un compteur 16 bits ; une demande d'interruption a lieu lorsqu'il y a dépassement de la capacité du compteur, càd. lorsque le compteur passe de FFFFh à 0000h ;
- le mode 2 : l'un des compteurs est utilisé comme compteur, l'autre comme registre dans lequel on peut écrire un nombre de 8 bits quelconque ; lorsqu'il y a dépassement de capacité du compteur, on charge dans celui-ci le contenu du registre; cette technique permet donc d'obtenir des demandes d'interruption régulièrement espacées si le compteur reçoit des impulsions d'horloge.

On peut choisir (bits appropriés de Tcon) d'appliquer à chaque compteur :

- soit des impulsions provenant d'une broche externe T0 ou T1 ;
- soit des impulsions provenant du circuit d'horloge, après passage par un diviseur de fréquence par 12.

Le fonctionnement des compteurs est contrôlé par :

- un bit du registre Tmod ;
- si désiré, l'état d'une broche externe (Int0 ou Int1).

Port série

Pour certaines applications (transfert de données à distance, commande de modems,...), il est utile de disposer en plus des ports d'E/S parallèles, d'un port d'entrée/sortie série ou sériel (*Serial Input/Output Port*).

Le principe des liaisons série est très simple : plutôt que d'envoyer simultanément 8 bits en parallèle sur 8 lignes, on envoie les 8 bits l'un après l'autre sur une seule ligne.

Le 8051 dispose d'un port série bidirectionnel utilisant une ligne pour l'envoi des données, TxD, et une ligne pour la réception des données, RxD.

Plusieurs modes de fonctionnement sont possibles. Le mode le plus utilisé emploie le temporisateur 1 en mode de rechargement automatique pour définir la cadence d'envoi des bits (*baud rate*). Les cadences standard vont de 300 bit/s à 19 200 bit/s.

Pour envoyer un caractère, il suffit de l'écrire dans le registre Sbuf d'émission ; cela enclenche automatiquement la procédure d'envoi du caractère :

- le premier bit s'appelle le bit Start, toujours à 0 ;
- il est suivi par les 8 bits de l'octet à transmettre ;
- on termine par le bit Stop, toujours à 1 ; la présence des bits Start et Stop assure qu'il y a toujours un flanc descendant en début de transmission d'un octet ; la ligne reste à 1 jusqu'à l'octet suivant ;
- lorsque Sbuf est vide, on déclenche une demande d'interruption ; le processeur enverra alors l'octet suivant, si nécessaire ;
- il est possible d'envoyer des mots de 9 bits plutôt que 8 ; ceci permet d'accoler aux 8 bits un bit de parité pour le contrôle des erreurs, ou de créer un petit réseau avec plusieurs processeurs.

Lors de la réception d'un caractère, la procédure est enclenchée par la détection d'un flanc descendant à la broche RxD. Les 8 ou 9 bits sont placés dans le registre Sbuf de réception (curieusement, on utilise le même nom et la même adresse, 99h, pour les deux registres Sbuf ; quand on écrit à cette adresse, l'octet est stocké dans le registre d'émission, quand on lit, c'est le contenu du registre de réception). Lorsque tous les bits sont présents, on déclenche une demande d'interruption. Il faut que le MC vienne lire l'octet reçu avant la fin de la réception de l'octet suivant, sinon le premier est perdu.

L'émetteur et le récepteur peuvent travailler simultanément. Les demandes d'interruption sont combinées. Donc, lorsque l'on envoie et reçoit simultanément des octets, le processeur doit déterminer lors de chaque demande d'interruption si elle émane de l'émetteur ou du récepteur. Cela se fait en allant lire les bits TI et RI du registre Scon (*Serial Port Control Register*).

Programmation

Outre l'assembleur, il est possible d'utiliser des compilateurs de langages de haut niveau.

Réduction de la consommation

De nombreuses applications ne nécessitent pas que le MC soit en permanence actif. Par exemple, un système d'acquisition de données peut parfaitement être "endormi" entre les moments où il fait les lectures de données, afin d'économiser l'énergie ; ceci est particulièrement intéressant lorsque les équipements sont alimentés par batterie.

Le 8051 en technologie NMOS ne possédait qu'un seul mode d'économie d'énergie, le mode ***Idle***. Dans ce mode, l'horloge n'est plus appliquée au microprocesseur, qui n'exécute donc plus d'instructions ; les registres internes préservent leurs informations ; le port série, la gestion des interruptions et les compteurs/temporisateurs fonctionnent normalement. On entre en mode *Idle* en mettant à 1 le bit IDL du registre Pcon (*Power Condition*).Le MC sort du mode *Idle* en cas d'interruption ou de Reset externe.

Le 8051 en technologie CMOS, la plus courante de nos jours, possède, outre le mode *Idle*, un mode ***Power Down***. Ici, c'est carrément l'oscillateur générant le signal d'horloge qui est bloqué. Toutes les fonctions sont bloquées. La différence avec une coupure d'alimentation, c'est que tous les registres préservent leurs informations, en particulier les ports d'E/S qui sont liés à des registres. On entre dans le mode *Power Down* en mettant à 1 le bit PD du registre Pcon. Le seul moyen de sortie du mode *Power Down*, c'est par un Reset externe, qui provoque aussi une réinitialisation du pointeur de programme. On recommence donc le programme à l'adresse 0000h et les SFR (dont les registres des ports) sont réinitialisés.

Famille du 8051

La famille du 8051 a connu un succès remarquable, malgré son jeu d'instructions passablement "exotique". C'est donc très naturellement qu'Intel d'abord, de nombreux autres constructeurs ensuite, ont mis au point des MC basés sur l'architecture 8051 mais dotés de divers perfectionnements : mémoire programme effaçable et reprogrammable (8751), troisième compteur/temporisateur (8052), capacités des mémoires ROM et RAM plus grandes, ports d'E/S plus nombreux, convertisseurs A/N et N/A intégrés, C/T supplémentaires (9 au total dans le 80251)...

Le 8052

Le 8052 est une version améliorée du 8051, le 8032 est une version améliorée du 8031, qui est un 8051 sans ROM interne.

Les principales améliorations apportées sont :

- la ROM interne comporte 8Ko
- la RAM interne comporte 256 octets au lieu de 128 ; toutefois, pour ne pas entraîner de conflit avec les SFR (registres de fonctions spéciales, qui occupent les adresses 80h à FFh), les 128 octets supérieurs de la RAM interne ne sont accessibles que par adressage indirect à registre, c'est-à-dire que l'adresse du registre auquel on veut accéder doit être placée dans un des registres de travail R0 ou R1 avant d'envoyer l'instruction utilisant le registre ;
- un troisième compteur/temporisateur 16 bits est prévu ; ce C/T possède trois modes de fonctionnement :

- Mode capture : le compteur compte les impulsions en provenance d'une entrée externe ou d'un diviseur de fréquence interne ; une impulsion appliquée à la broche T2EX copie l'état du compteur au moment de cette impulsion dans deux SFR (RCAP2H et RCAP2L) ; une demande d'interruption est générée ; le compteur continue à compter normalement.

- Mode rechargement automatique : le contenu des SFR RCAP2H et RCAP2L est copié dans le compteur soit automatiquement, lors d'un dépassement de capacité du compteur, soit lorsqu'une impulsion externe est appliquée à la broche T2EX.

- Mode horloge de transmission : ce mode est destiné à faciliter la réalisation de l'interface série.

Le 8052 Basic

Il s'agit d'une version spéciale du 8052. La mémoire ROM interne de 8Ko est occupée par un interpréteur Basic. Il est donc possible, en utilisant un terminal et l'interface série du 8052, de travailler en mode interactif, en envoyant au 8052 des commandes en langage Basic à partir du terminal.

Le Basic 8052 :

- traite des nombres en code DCB (décimal codé binaire) de 8 chiffres + exposant ;
- permet, outre les 4 opérations arithmétiques de base, de calculer la racine carrée d'un nombre, ainsi que son sinus, cosinus, tangente, arc tangente, logarithme et exponentielle ;
- permet la commande directe des périphériques ;
- opère la conversion décimal-hexadécimal et inversement ;
- permet la gestion des ressources du microcontrôleur : compteurs, interface série, interruptions...
- génère les signaux nécessaires pour programmer des EPROM, EEPROM...;
- simule une horloge temps réel.

Liens externes

- Intel (http://www.intel.com/design/mcs51)
- Philips (http://www.nxp.com/products/microcontrollers/80c51/index.html)
- Analog device (http://www.analog.com/en/subCat/0,2879,762%255F%255F0%255F%255F0%255F,00.html)
- Maxim (http://www.maxim-ic.com/products/microcontrollers/8051)
- Tutoriel (http://www.ikalogic.com/tut_8051_1.php) micro-contrôleur 8051