Le Microcontrôleur 8051/8052

A. Oumnad

I Introduction

Bien que le 8051 soit un circuit assez ancien, il reste aujourd'hui un des microcontrôleurs les plus populaire.

Fin 1979, INTEL commercialise la famille de microcontrôleurs MCS 51 qui correspond au départ à trois types de microcontrôleurs ; le 8051 (à mémoire ROM), le 8751 (à mémoire EPROM) et le 8031 (ROMLESS).

Le succès de la famille 8051 a amené la fabrication de ce microcontrôleur et de ses dérivés par de nombreux constructeurs de CI : PHILIPS, DALLAS, ATMEL, SIEMENS pour ne citer que les plus importants. On trouve aussi des cœur de 8051 (en VHDL) vendu en propriété industrielle. Tous ses produits sont compatibles, avec des vitesses d'horloges différentes, des nouvelles fonctionnalités (contrôleur I2C, CAN, watchdog ...). De nouvelle forme de programmation (programmation ISP pour ATMEL), taille de mémoires plus grandes, nombre de ports E/S plus grands. Le 8051 et ses produits dérivés reste le microcontrôleur 8 bits le plus vendu dans le monde.

Il est alors important de s'intéresser à l'architecture du 8051 car elle commune à tous les microcontrôleurs de la famille.

II Présentation de la famille MCS 51 :

Les caractéristiques principales de quelques échantillons de la famille MCS 51 sont données dans le tableau ci-dessous :

DEVICE	mémoire de programme	RAM de données	vitesse	ports d'E/S	timers/ compteurs	UART
8031	ROMLESS	128 o	12 MHz	4 x 8 bits	2	1
8051	4K ROM	128 o	12 MHz	4 x 8 bits	2	1
8751	4K EPROM	128 o	12 MHz	4 x 8 bits	2	1
8032	ROMLESS	256 o	12 MHz	4 x 8 bits	3	1
8052	8 K ROM	256 o	16 MHz	4 x 8 bits	3	1
8752	8 K EPROM	256 o	20 MHz	4 x 8 bits	3	1

Tableau II.1 : 8051 et 8052 avec leurs version ROMLESS et EEPROM

II.1 Caractéristiques principales du 8051

Le microcontrôleur possède les caractéristiques suivantes:

- un CPU à 8 bits conçu pour la commande d'applications diverses,
- 32 entrées/sorties bidirectionnelles qui peuvent être adressées individuellement réparties en 4 ports : P0, P1, P2, P3.
- 128 octets de RAM interne à utilisation générale
- 21 registres spécialisés
- un port série en full duplex
- 5 sources d'interruptions avec 2 niveaux de priorité
- 2 Compteurs/Timers sur 16 bits T0 et T1 fonctionnant suivant 4 modes
- un oscillateur interne nécessitant un quartz externe : la fréquence d'oscillation maximale admise est de 12 MHz
- Adressage de 64 Ko de mémoire de données
- Adressage de 64 Ko de mémoire de programme
- un jeu d'instructions assez développé

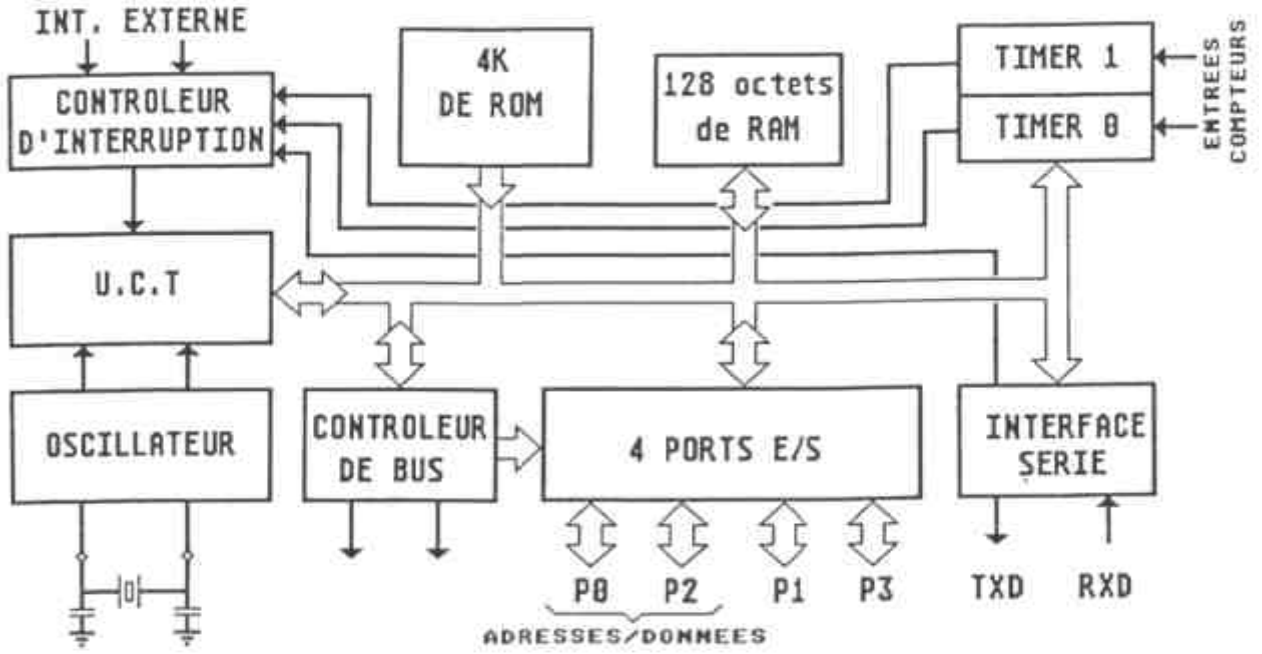


Figure II.1 : Architecture simplifiée du 8051

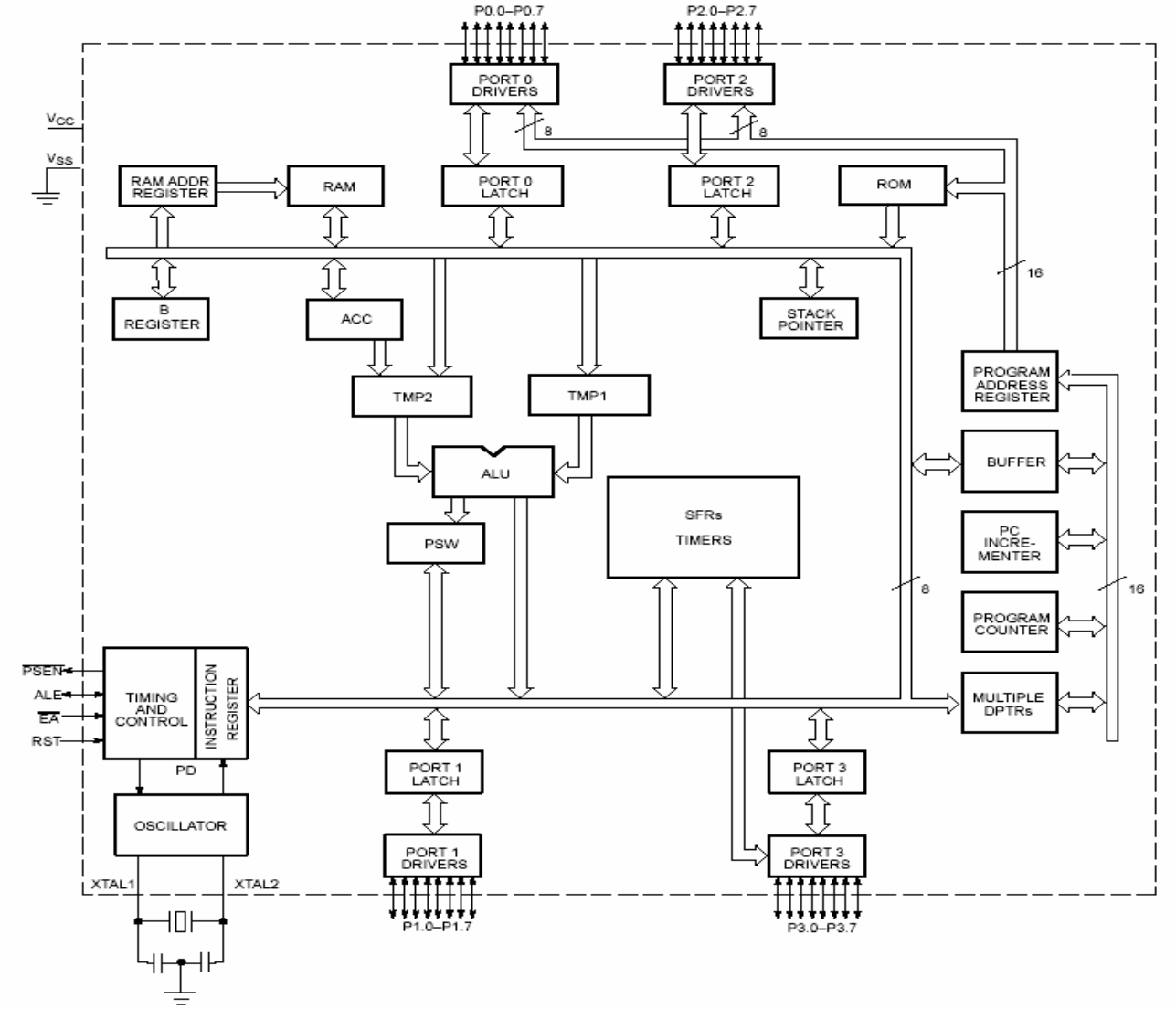
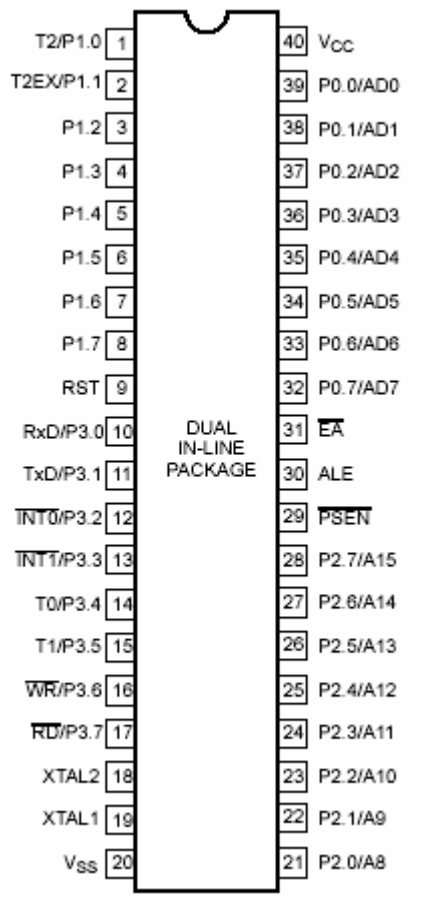


Figure II.2 : Architecture détaillée du 8051

II.2 Brochage du 8051



II.2.1 Le port P0

C'est un port 8 bits bidirectionnel à usage général à sorties drain ouverts. Il a la fonction secondaire de port multiplexé transportant les 8 bits inférieurs des bus de données et adresse permettant d'accéder à une mémoire externe de type RAM de données ou EEPROM programme, dans ce cas les sorties sont dotées de résistances de pull-up internes.

II.2.2 Le port P2

C'est un port 8 bits bidirectionnel à usage général avec des résistances de pull-up internes avec la fonction secondaire de port multiplexé transportant les 8 bits supérieurs des bus de données et d'adresse.

II.2.3 Le port P1

C'est un port 8 bits bidirectionnel à usage général avec résistances de pull-up internes avec les fonctions secondaires (8052) :

P1.0 sert aussi comme horloge externe pour le *Timer* 2,
P1.1 sert aussi comme entrée de control du *Timer* 2,

II.2.4 Le port P3

C'est un port 8 bits bidirectionnel à usage général avec résistances de pull-up internes avec les fonctions secondaires :

P3.0 = RxD : Entrée de l'interface série
P3.1 = TxD : Sortie de l'interface série
P3.2 = /INT0 : entrée d'interruption
P3.3 = /INT1 : entrée d'interruption
P3.4 = T0 : entrée horloge du *timer* 0
P3.5 = T1 : entrée horloge du *timer* 1
P3.6 = /WR : sortie écriture de la mémoire externe
P3.7 = /RD : sortie lecture de la mémoire externe

II.2.5 Les autre E/S

- **/EA** : (**E**xternal **A**ccess) si EA=0, les instructions sont recherchées dans la mémoire programme externe.
- **RST** : Entrée d'initialisation. Un état haut pendant deux cycles machines sur cette broche entraîne une initialision du microcontrôleur.
- **/PSEN** : (**P**rogramm **S**tore **EN**able) passe à 0 lorsque le micro va rechercher une instruction en mémoire programme externe.
- **ALE** : *(Adress Latch Enable)* prévue pour commander le démultiplexage du port P0.
  - ALE = 1, P0 transporte la partie basse du bus d'adresse : A0 à A7
  - ALE = 0, P0 sert de bus de donnée
- **XTAL1 et XTAL2** : Placer le quartz entre ces deux broches avec deux condensateurs de 22pF entre ces deux broches et la masse

II.3 Organisation de la mémoire

Les microcontrôleurs de la famille 8051 manipulent plusieurs types de mémoire comme indiqué sur la Figure II.3. On distingue :

- La mémoire interne *(on chip memory)*, constituée d'une RAM de données, de registre de contrôle *(SFR : Special Function Registers)*, et d'une mémoire programme qui est en général une ROM ou de préférence une EEPROM,
- La mémoire externe constituée d'une RAM de données et d'une mémoire programme en général de type EEPROM

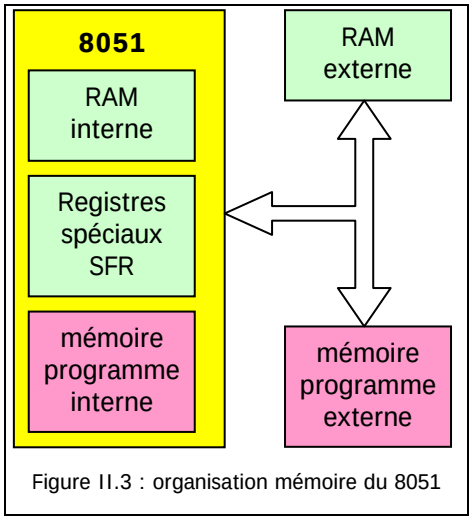


Figure II.3 : organisation mémoire du 8051

II.3.1 La mémoire programme

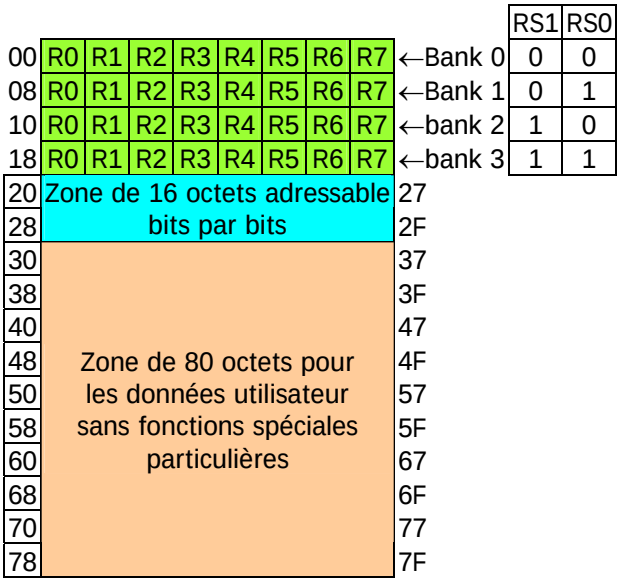
Sur le 8051, il y a une mémoire programme interne de type ROM de 4ko, il peut toutefois adresser jusqu'à 64 ko de mémoire externe. La ligne EA permet de distinguer l'accès à la mémoire interne ou externe. Le 8031 n'a pas de mémoire programme interne.

II.3.2 La mémoire RAM

Le 8051 dispose 128 octets de RAM interne et peut adresse jusqu'à 64 ko de RAM externe. la RAM interne est bien plus rapide que la RAM externe, par exemple : pour incrémenter une position RAM interne il faut un cycle machine, alors que pour incrémenter une position RAM externe, il faut 7 cycles machine, dans ce cas, la RAM externe est 7 fois plus lente que la RAM interne.

II.3.3 Organisation de la RAM interne

Le 8051 possède 128 octets de RAM interne adressable de 00 à 7F. Cette mémoire est organisée en 3 zones :



Les banks de registres :

Les 32 premier octets sont utilisée soit comme mémoire pour les données utilisateurs soit comme registres de travail appelés aussi registres universels. Les registres portent les noms de R0 à R7. Si on observe la figure, on s'aperçoit que les 8 registres sont reproduits 4 fois. Cela signifie qu'un registre Rx correspond à 4 positions mémoire différentes, comment faire alors la distinction. Chaque rangée de 8 octets est appelée *bank*, et pour passer d'un *bank* à l'autre il faut positionner les deux bits 3 et 4 (RS0, RS1) du registre de contrôle PSW. Au RESET c'est le *bank0* qui est sélectionné.

Si on veut écrire la donnée 34 dans la position d'adresse 05, les deux instructions suivantes sont équivalentes :

MOV 05,#34 ou MOV R5,#34

Si on veut écrire la donnée 34 dans la position d'adresse 12, les deux instructions suivantes sont équivalentes :

MOV PSW,xxx01xxxb

MOV 0Ch,#34 ou MOV R4,#34

La mémoire adressable par bits

Les 16 octets suivants constituent une zone adressable bit par bit. (*L'adressage par octet reste évidemment valable)* à l'aide d'instructions du genre **SETB** ou **CLR.** La zone contient 128 bits adressables de 0 à 7Fh (0 à 127).

Si on désire positionner à 1 le LSB de la position mémoire 22h, on peut écrire :

SETB 10h ou ORL 22h,#01h

Adr	b7	b6	b5	b4	b3	b2	b1	b0
20*h*	07	06	05	04	03	02	01	00
21*h*	0F	0E	0D	0C	0B	0A	09	08
22*h*	17	16	15	14	13	12	11	10
23*h*	1F	1E	1D	1C	1B	1A	19	18
24*h*	27	26	25	24	23	22	21	20
25*h*	2F	2E	2D	2C	2B	2A	29	28
26*h*	37	36	35	34	33	32	31	30
27*h*	3F	3E	3D	3C	3B	3A	39	38
28*h*	47	46	45	44	43	42	41	40
29*h*	4F	4E	4D	4C	4B	4A	49	48
2A*h*	57	56	55	54	53	52	51	50
2B*h*	5F	5E	5D	5C	5B	5A	59	58
2C*h*	67	66	65	64	63	62	61	60
2D*h*	6F	6E	6D	6C	6B	6A	69	68
2E*h*	77	76	75	74	73	72	71	70
2F*h*	7F	7E	7D	7C	7B	7A	79	78

La pile :

A la mise sous tension (RESET), le processeur place le début de la pile à l'adresse 07h (SP=07h), ce qui signifie que l'écriture dans la pile commencera à l'adresse 08h. Dans le cas ou l'on désire utiliser les *banks* de registres, il est préférable de placer la pile dans un autre endroit en changeant la valeur su SP. (60h par exemple)

II.3.4 Les registres spéciaux SFR

Le 8051 possède 21 registres spéciaux nécessaire au fonctionnement et à la configuration du microcontrôleur, on y trouve aussi des registres pour accéder aux ports d'E/S et aux timers.

Les registres sont accessibles comme des positions mémoire sauf que leurs adresses se situent dans l'espace 80h – FFh. Certains SFRs sot adressables bit par bit (Tableau II.3)

80	**P0**	**SP**	**DPL**	**DPH**				**PCON**
88	**TCON**	**TMOD**	**TL0**	**TL1**	**TH0**	**TH1**		
90	**P1**							
98	**SCON**	**SBUF**						
A0	**P2**							
A8	**IE**							
B0	**P3**							
B8	**IP**							
C0								
C8	**T2CON**			**RCAP2L**	**RCAP2L**	**TL2**	**TH2**	
D0	**PSW**							
D8								
E0	**ACC**							
E8								
F0	**B**							
F8								

87	8F	97	9F	A7	AF	B7	BF	C7	CF	D7	DF	E7	EF	F7	FF

Tableau II.2 : cartographie des registres SFR 8051

Registres de travail :

A : Accumulateur
B : Accumulateur auxiliaire
R0 à R7 : Registres à usage général et adressage

Ports d'E/S :

P0 : Port 0
P1 : Port 1
P2 : Port 0
P3 : Port 1

Port série :

SCON : Configuration du port série
SBUF : Lecture/Ecriture dans le port série

Timers :

TCON : Configuration des Timers 0 et 1
T2CON : Configuration du Timer 2
TMOD : Configuration des modes de fonctionnement des Timers
TH0 : Octet haut de TIMER0
TL0 : Octet bas de TIMER0
TH1 : Octet haut de TIMER1
TL1 : Octet bas de TIMER1
TH2 : Octet haut de TIMER2
TL2 : Octet bas de TIMER2
RCAP2H : Octet haut du registre de chargement/capture de TIMER2
RCAP2L : Octet bas du registre de chargement/capture de TIMER2

Interruptions :

IE : Validation d'interruption
IP : Priorité des interruptions

Pointeurs d'adressage :

DPH : Octet haut du registre DPTR (adressage mémoire externe)
DPL : Octet bas du registre DPTR (adressage mémoire externe)
SP : Pointeur de pile

Configuration générale :

PCON : Contrôle de consommation
PSW : Registre d'état, sélection de bank

Symbole	Fonction	Adr.	Adresse au niveau du bit								Etat initial
P0	Port P0	80h	87	86	85	84	83	82	81	80	11111111
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
SP	Pointeur de pile	81h									00000111
DPL	poids faible de DPTR	82h									00000000
DPH	Poids fort de DPTR	83h									00000000
PCON	Mode de consommation	87h	--------	--	--	--	-----	-----	----	-----	0xxxxxxx
			SMOD	--	--	--	GF1	GF0	PD	IDL	
TCON	Contrôle de T1 et T2	88h	8F	8E	8D	8C	8B	8A	89	88	00000000
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TMOD	Modes pour T0 et T1	89h	--------	-----	----	----	--------	-----	----	----	00000000
			GATE	C/T	M1	M0	GATE	C/T	M1	M0	
TL0	poids faible du T0	8Ah									00000000
TL1	poids faible du T1	8Bh									00000000
TH0	Poids fort du Timer 0	8Ch									00000000
TH1	Poids fort du Timer 1	8Dh									00000000
P1	Port P1	90h	97	96	95	94	93	92	91	90	11111111
			---	---	---	---	---	---	T2EX*	T2*	
SCON	Contrôle du port série	98h	9F	9E	9D	9C	9B	9A	99	98	00000000
			SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
SBUF	Données du port série	99h									00000000
P2	Port P2	A0h	A7	A6	A5	A4	A3	A2	A1	A0	11111111
			A15	A14	A13	A12	A11	A10	A9	A8	
IE	Validation des int.	A8h	AF	AE	AD	AC	AB	AA	A9	A8	0x000000
			EA	---	ET2	ES	ET1	EX1	ET0	EX0	
P3	Port P3	B0h	B7	B6	B5	B4	B3	B2	B1	B0	11111111
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	
IP	priorité des interruptions	B8h	BF	BE	BD	BC	BB	BA	B9	B8	xx000000
			---	---	PT2	PS	PT1	PX1	PT0	PX0	
T2CON*	Contrôle de T2	C8h	CF	CE	CD	CC	CB	CA	C9	C8	00000000
			TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	
RCAP2L*	Capture/rech du T2(bas)	CBh									00000000
RCAP2H*	Capture/rech du T2 (haut)	CCh									00000000
TL2*	poids faible du Timer 2	CDh									00000000
TH2*	Poids fort du Timer 2	CEh									00000000
PSW	registre d'état et sélection de *bank*	D0h	D7	D6	D5	D4	D3	D2	D1	D0	00000000
			CY	AC	F0	RS1	RS0	0V	---	P	
ACC	Accumulateur A	E0h	E7	E6	E5	E4	E3	E2	E1	E0	00000000
B	Usage général Obligatoire pour mul. et div.	F0h	F7	F6	F5	F4	F3	F2	F1	F0	00000000

Tableau II.3 : cartographie détaillée des registres SFR du 8051/8052

Accumulateur A (E0h)

Il est référencé par ACC sur le tableau, cependant dans les instructions on le désigne par A

Accumulateur B (F0h)

Il est utilisé tout particulièrement pour l'exécution des multiplications et divisions. Mais il peut être considéré aussi comme un registre quelconque.

P0 : Port 0 (80h)

Registre d'accès au port P0

87	86	85	84	83	82	81	80
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

P1 : Port 1 (90h)

Registre d'accès au port P1

97	96	95	94	93	92	91	90
---	---	---	---	---	---	T2EX*	T2*

P2 : Port 2 (A0h)

Registre d'accès au port P2

A7	A6	A5	A4	A3	A2	A1	A0
A15	A14	A13	A12	A11	A10	A9	A8

P3 : Port 3 (B0h)

Registre d'accès au port P3

B7	B6	B5	B4	B3	B2	B1	B0
RD	WR	T1	T0	INT1	INT0	TxD	RxD

PSW : Program Status Word (D0h)

PSW est un registre d'état qui contient les drapeaux positionnés après les instruction ainsi que les bits de sélection de banks

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	0V	---	P

P : parité de l'accumulateur = 1 si le nombre de 1 dans l'accu est impair (odd)
OV : Overflow
RS0, RS1 : pour sélectionner un *bank* de registres
FO : drapeau à usage général
AC : Carry auxiliaire pour les calculs en BCD
CY : Carry

SP(81h)

Pointeur de pile. Lors d'un reset le pointeur est initialisé à 07H, et par conséquent la pile débute à l'adresse 08H. L'utilisateur peut changer la valeur de SP pour placer la pile dans une autre zone de la RAM,

DPL, DPH (82h,83h)

Partie basse et partie haute du pointeur de donnée DPTR utilisé avec certaines instructions d'accès à la RAM externe

SBUF (99h)

C'est le *buffer* d'entrée sortie du port série. Tout octet écrit dans SBUF et transmis en série sur TxD et tout octet reçu sur RxD peut être lu dans SBUF.

TL0, TH0 (90h, 92h)

Partie basse et haute du registre double qui permet d'accéder au *Timer* T0

TL1, TH1 (91h, 93h)

Partie basse et haute du registre double qui permet d'accéder au *Timer* T1

SCON : Serial Control (98h)

Ce registre adressable par bits sert à la configuration du port série

9F	9E	9D	9C	9B	9A	99	98
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Nous reviendrons sur le fonctionnement du port série dans un prochain paragraphe

TCON : Timer Control (88h)

Ce registre adressable par bits permet le contrôle des timers T0 et T1. Il contient aussi certains bits relatifs à l'utilisation des interruptions externes.

8F	8E	8D	8C	8B	8A	89	88
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TMOD : Timer mode (89h)

Ce registre permet de définir le mode de fonctionnement des timers T0 et T1

GATE	C/T	M1	M0	GATE	C/T	M1	M0

IE : Interrupt Enable (A8h)

Ce registre permet la validation ou l'interdiction des interruptions. Les premiers 7 bits permettent de valider les interruptions individuellement alors que le 8ème bit permet une validation globale de toutes les interruptions.

AF	AE	AD	AC	AB	AA	A9	A8
EA	---	ET2	ES	ET1	EX1	ET0	EX0

IP : Interrupt Priority (B8h)

Ce registre permet de définir la priorité relative pour chaque interruption. Sur le 8051 une interruption peut avoir deux niveaux de priorité, basse (0) et Haute (0). Une interruption ne peut être interrompue que par une interruption de priorité supérieure.

BF	BE	BD	BC	BB	BA	B9	B8
---	---	PT2	PS	PT1	PX1	PT0	PX0

PCON : Power Control (87h)

Registre de contrôle de la consommation du 8051

TL2, TH2 (CDh, CEh) (8052)

Partie basse et haute du registre double qui permet d'accéder au *Timer* T2

RCAP2L, RCAP2H (CBh, CCh) (8052)

Partie basse et partie haute du registre de capture qui permet sous certaine condition de faire une recopie du *timer* 2

T2CON : Timer Control (C8h)

Ce registre adressable par bits permet le contrôle de TIMER2.

CF	CE	CD	CC	CB	CA	C9	C8
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

III Modes d'adressage

III.1 Adressage Immédiat

Ce n'est pas réellement un adressage car la donnée constituant l'opérande est précisée dans l'instruction. Dans l'écriture en assembleur, on place un signe # devant la donnée pour la différentier d'une adresse.

Exemples:

MOV A,#40h : Chargement de l'octet 40h dans l'accumulateur
MOV A,#40 : charger la valeur 40 décimal dans l'accumulateur
MOV A,#11001101B : charger la valeur binaire dans l'accumulateur
MOV P1,#80H : envoie la valeur 80h sur le port P1
MOV 40h,#5AH : Charger la valeur 5Ah dans la case mémoire d'adresse 40h

III.2 Adressage Registre

L'opérande est contenu dans un registre
ADD A,R4
INC A
INC R3

III.3 Adressage direct

C'est l'adresse de l'opérande qui est précisée dans l'instruction. L'adresse n'a que 8 bits, Seule la RAM interne et les SFRs peuvent être adressés de cette manière. Il n'existe pas d'adressage direct pour la RAM externe, il faut passer par le registre DPTR.

Exemple:

MOV A,45h : Charger l'accumulateur avec le contenu de la case mémoire 45h
MOV R0,30h : Charger le registre R0 avec le contenu de la case mémoire 30h
MOV A,P1 : Charger l'accumulateur avec le mot en entrée sur le port P1
ANL A,78 : Une opération ET logique est effectuée entre le contenu de A et le contenu de la case mémoire 78 (4Eh)

III.4 Adressage Indirect (indexé)

L'instruction spécifie un registre qui contient l'adresse de l'opérande. Les mémoires internes et externes sont accessibles par ce type d'adressage (sauf les registres SFR). Dans le cas d'un adressage sur 8 bits le registre d'index peut être l'un des 2 registres R0 ou R1 choisi dans le *bank* actif. Dans le cas de l'adressage 16 bits, qui ne concerne que la mémoire externe, le registre d'index est le DPTR (16 bits). L'adressage indirect est désigné dans l'assembleur par le signe @.

Exemple:

MOV A,@R0 : Charger l'accumulateur avec le contenu de la case dont l'adresse est contenue dans registre R0 du *bank* actif
MOV @R1,P1 : Transférer le contenu du port P1 dans la case mémoire dont l'adresse est le contenu du registre R1 du *bank* actif.

ANL A,@R0 : ET logique entre l'accumulateur et la case mémoire dont l'adresse est le contenu du registre R0 du *bank* actif

MOVX @DPTR,A : Charge la mémoire dont l'adresse est contenue dans DPTR dans l'accumulateur. Le X du mnémonique indique qu'il s'agit d'une mémoire extérieure.

L'utilisation des registres d'index R0 et R1 avec la mémoire externe peut s'avérer assez souple, mais elle permet seulement l'adressage à l'intérieur d'une page mémoire de 256 octets. L'adresse de la page doit être précisée à l'aide du port P2.

Les 3 instructions suivantes permettent de charger l'accumulateur avec le contenu de la case mémoire externe d'adresse 43A0h

MOV P2,#43h
MOV R0,#A0h
MOVX A,@R0

III.5 Adressage de Bits

Le 8051 possède un processeur de bits qui travaille sur des bits individuels. Pour ce processeur l'Accumulateur est C. Les bits de certains registres sont ainsi accessibles par une adresse sur 1 octet.

D'abord les bits de la zone en début de RAM interne (16 octets de 20H à 2FH). Ces 128 bits qui commencent Bit 0 de la case 20h et se terminent au bit 7 de la case 2F ont des adresses qui vont de 00 à 7Fh.

Sont également accessibles de cette façon les bits des cases du SFR dont l'adresse en hexadécimal se termine par 0 ou 8, 80H 88H 90H 98H etc...

L'un des opérandes est toujours le carry C :

MOV C,45h : Chargement de C avec le bit d'adresse 45h
MOV P1.3,C : Le contenu de C est transféré dans le bit 3 du port P1
MOV RS0,C : Le bit C est placé dans le bit RS0 du Registre PSW
CLR C : mise à zéro de C
CLR P3,6 : mise à zéro du bit 6 de P3
SETB RS1 : mise à 1 de RS1
CPL P1.1 : Inversion du bit 1 de P1
SETB P3.2 : Mise à 1 du bit 2 du port 3

IV Jeux d'instruction du 8051

Signification des symboles utilisés:

Rn	Un des registres actifs R0 à R7
direct	Adresse d'un octet de RAM interne, d'un port, ou SFR
@Ri	Adressage indirect par registre R0 ou R1
#data	Donnée 8 bits
#data16	Donnée 16 bits
bit	Adresse au niveau du bit
rel	déplacement 8 bits signé relatif au PC : de -128 à +127
addr11	Adresse limitée au bloc de 2Ko dans lequel figure l'instruction
addr16	Adresse sur l'espace de 64Ko

IV.1 Instructions de transfert de données

Les transferts affectant l'accumulateur s'effectuent en 1µS, les autres en 2µs.

MOV : déplacement de donnée dans la mémoire interne

MOV Dest,Source : recopie Source dans Dest, aucun drapeau n'est positionné.

Notons que l'on peut transférer des données directement d'une case mémoire interne dans une autre sans passer par l'accumulateur.

Les instruction du genre MOV Rn,Rn ne sont pas autorisé. Pour réaliser M0V R2,R4 avec le *bank* 0 actif, il faut faire MOV 02,04. Si c'est le *bank* 1 qui est actif, pour réaliser MOV R2,R4 il faut faire MOV 0Ah,0Ch

Syntaxe	Code	Octets	cycles	Flags
MOV A, Rn	E8+n	1	1	-
MOV A, direct	E5	2	1	-
MOV A, @Ri	E6+i	1	1	-
MOV A, #data	74	2	1	-
MOV Rn, A	F8+n	1	1	-
MOV Rn, direct	A8+n	2	2	-
MOV Rn, #data	78+n	2	1	-
MOV direct, A	F5	2	1	-
MOV direct, Rn	88+n	2	2	-
MOV direct, direct	85	3	2	-
MOV direct, @Ri	86+i	2	2	-
MOV direct, #data	75	3	2	-
MOV @Ri, A	F2+i	1	1	-
MOV @Ri, direct	A6+i	2	2	-
MOV @Ri, #data	76+i	2	1	-
MOV DPTR, #data16	90	3	2	-
MOV bit, C	92	2	2	-
MOV C, bit	A2	2	1	C

PUSH et POP

Sauvegarder ou récupérer une valeur dans la pile.
PUSH incrémente SP puis écrit l'opérande dans la case pointé par ce dernier
POP lit la valeur pointée par SP avant de décrémenter ce dernier

XCH

XCH A,<byte> : échange les données de l'accumulateur A et de l'octet adressé.

XCHD

XCHD A,@Ri : échange les quartets de poids faible entre l'accu A et l'octet adressé.

Syntaxe	Code	Octets	cycles
PUSH direct	C0	2	2
POP direct	D0	2	2
XCH A, Rn	C8+n	1	1
XCH A, direct	C5	2	1
XCH A, @Ri	C6+i	1	1
XCHD A, @Ri	D6+i	1	1

MOVX

MOVX <dest>, <source> permet la lecture ou l'écriture d'un octet en RAM externe en passant toujours par l'accumulateur. L'adressage indirect est utilisé

MOVC

MOVC A, @A+<base-reg> permet de lire un octet dans la mémoire programme.

Syntaxe	Code	Octets	cycles
MOVX A,@Ri	E2+i	1	2
MOVX A,@DPTR	E0	1	2
MOVX @Ri,A	F2+i	1	2
MOVX @DPTR,A	F0	1	2
MOVC A,@A+DPTR	93	1	2
MOVC A,@A+PC	83	1	2

IV.2 Instructions arithmétiques

Syntaxe	Code	Octets	cycles	Flags
ADD A, Rn	28+n	1	1	C, AC, O
ADD A, direct	25	2	1	C, AC, O
ADD A, @Ri	26+i	1	1	C, AC, O
ADD A, #data	24	2	1	C, AC, O
ADDC A, Rn	38+n	1	1	C, AC, O
ADDC A, direct	35	2	1	C, AC, O
ADDC A, @Ri	36+i	1	1	C, AC, O
ADDC A, #data	34	2	1	C, AC, O
SUBB A, Rn	98+n	1	1	C, AC, O
SUBB A, direct	95	2	1	C, AC, O
SUBB A, @Ri	96+i	1	1	C, AC, O
SUBB A, #data	94	2	1	C, AC, O
INC @Ri	06+i	1	1	-
INC A	04	1	1	-
INC direct	05	2	1	-
INC Rn	08+n	1	1	-
INC DPTR	A3	1	2	-
DEC @Ri	16+i	1	1	-
DEC A	14	1	1	-
DEC direct	15	2	1	-
DEC Rn	18+n	1	1	-
MUL AB	A4	1	4	C, O
DIV AB	84	1	4	C, O
DA A	D4	1	1	C

ADD

ADD A, <byte> : additionne un octet avec l'accu A, résultat dans A.

ADDC

ADDC A, <byte> : additionne un octet, l'accu A et la retenue, résultat dans A.

SUBB

SUBB A, <byte> : soustrait un octet et la retenue du contenu de A, résultat dans A.

INC

INC <byte> : incrémente un octet ou DPTR.

DEC

DEC <byte> : décrémente un octet.

MULL

MUL AB : multiplie l'accu A et l'accu B, résultat : octet faible dans A et octet fort dans B

DIV

DIV AB : divise le contenu de A par le contenu de B, quotient dans A et reste dans B.

DA

DA A : ajustement décimal de A.

IV.3 Instructions logiques et booléennes

ANL

ANL <dest>, <source> : réalise un ET logique entre source et dest, résultat dans dest.

ORL

ORL <dest>, <source> : réalise un OU logique entre source et dest, résultat dans dest.

XRL

XRL <dest>, <source> : réalise un OU exlusif entre source et dest, résultat dans dest.

CLR

met l'accumulateur ou un bit à 0

CPL

Complémente l'accumulateur ou un bit

RL, RLC, RR, RRC, SWAP

RL A : rotation vers la gauche du contenu de A
RLC A : rotation vers la gauche du contenu de A à travers la retenue
RR A : rotation vers la droite du contenu de A
RRC A : rotation vers la droite du contenu de A à travers la retenue
SWAP A : échange le quartet de poids faible avec celui de poids fort de A

SETB

SETB <bit> : met un bit à 1

ANL A, #data	54	2	1	-
ANL A, @Ri	56+i	1	1	-
ANL A, direct	55	2	1	-
ANL A, Rn	58+n	1	1	-
ANL direct, #data	53	3	2	-
ANL direct, A	52	2	1	-
ANL C, /bit	B0	2	2	C
ANL C, bit	82	2	2	C
ORL A, #data	44	2	1	-
ORL A,@Ri	46+i	1	1	-
ORL A, direct	45	2	1	-
ORL A, Rn	48+n	1	1	-
ORL direct, #data	43	3	2	-
ORL direct, A	42	2	1	-
ORL C, /bit	A0	2	2	C
ORL C, bit	72	2	2	C
XRL A, #data	64	2	1	-
XRL A, @Ri	66+i	1	1	-
XRL A, direct	65	2	1	-
XRL A, Rn	68+n	1	1	-
XRL direct, #data	63	3	2	-
XRL direct, A	62	2	1	-
CLR A	E4	1	1	-
CLR bit	C2	2	1	-
CLR C	C3	1	1	C=0
CPL A	F4	1	1	-
CPL bit	B2	2	1	-
CPL C	B3	1	1	C
RL A	23	1	1	-
RLC A	33	1	1	C
RR A	03	1	1	-
RRC A	13	1	1	C
SWAP A	C4	1	1	-
SETB bit	D2	2	1	-
SETB C	D3	1	1	-

IV.4 Les instruction de branchement

ACALL

ACALL addr11 : réalise un saut absolu inconditionnel vers un sous programme commençant à l'adresse addr11. Les 11 bits de addr11 sont substitués au 11 bits de poids faible du PC, ce qui signifie que la destination reste dans le même bloc de 2K que l'instruction courante.

Le codage se fait d'une façon particulière : les 3 bits de poids fort de addr11 complètent le code Opération dont les 5 bits bas sont égaux 11h

Ad_{10} Ad_9 Ad_8	Code opération	Ad_7 Ad_6 Ad_5 Ad_4 Ad_3 Ad_2 Ad_1 Ad_0

Si on note Ad_{10} Ad_9 Ad_8 = page = P et Ad_7 Ad_6 Ad_5 Ad_4 Ad_3 Ad_2 Ad_1 Ad_0 = Addr8
On obtient le code instruction :

11h+20hxP Addr8 (P variant entre 0 et 7)

AJMP

AJMP addr11 : réalise un saut absolu inconditionnel vers l'instruction d'adresse addr11. La différence avec ACALL est que ici on ne saute pas vers un sous programme, il n'est pas nécessaire d'empiler le PC pour pouvoir retourner après l'exécution du sous programme. Le codage se fait de la même façon que ACALL sauf que CO_L = 01h :
01h+20hxP Addr8 (P variant entre 0 et 7)

LCALL

LCALL addr16 : réalise un saut long absolu inconditionnel vers un sous programme commençant à l'adresse addr16. Pour repérer la position de retour, le PC est incrémenté de 3 puis empilé. Il sera dépilé au retour du sous programme.

LJMP

LJMP addr16 : réalise un saut long absolu inconditionnel vers la position d'adresse addr16.

SJMP

SJMP rel : réalise un saut court relatif au PC. On peut donc sauter dans l'intervalle :
[PC-128, PC+127]

JMP

JMP @A+DPTR : réalise un saut long absolu inconditionnel vers la position d'adresse A+DPTR. (Adressage indirect)

Syntaxe	Code instruction	Octets	cycles
ACALL adr11	11h+20hxP Addr8	2	2
AJMP addr11	01h+20hxP Addr8	2	2
LCALL addr16	12h addr16	3	2
LJMP addr16	02h addr16	3	2
SJMP rel	80h rel	2	2
JMP @A+DPTR	73h	1	2

IV.5 Les instruction de branchement conditionnels

- JZ rel : saut si A=0
- JNZ rel : saut si A<>0
- JC rel : saut si retenue à 1
- JNC rel : saut si retenue à 0
- JB bit,rel : saut si bit à 1
- JNB bit,rel : saut si bit à 0
- JBC bit,rel : saut si le bit est à 1 et mise à zero de celui-ci

- CJNE <byte1>, <byte2>, <rel> : saut si byte1 et byte2 sont différents
- DJNZ <byte>, rel : décrémente byte et saut si résultat différent de 0

JZ rel	60	2	2
JNZ rel	70	2	2
JC rel	40	2	2
JNC rel	50	2	2
JB bit, rel	20	3	2
JNB bit, rel	30	3	2
JBC bit,rel	10	3	2
CJNE @Ri, #data, rel	B6+i	3	2
CJNE A, #data, rel	B4	3	2
CJNE A, direct, rel	B5	3	2
CJNE Rn, #data, rel	E8+n	3	2
DJNZ direct, rel	D5	3	2
DJNZ Rn, rel	D8+n	2	2

IV.6 Instruction Diverses

RET : Retour de sous programme

RETI : Retour d'interruption

NOP : No Operation

Syntaxe	CO	Octets	cycles
RET	22h	1	2
IRET	32h	1	2
NOP	00	1	1

Syntaxe	Description	Octets	cycles	Flags
Instruction de Transfert				
MOV A, Rn	Copier le contenu du registre dans l'Acc.	1	1	-
MOV A, direct	Copier le contenu de la case adressée dans l'Acc.	2	1	-
MOV A, @Ri	Copier le contenu de la case adressée par Ri dans l'Acc.	1	1	-
MOV A, #data	Copier la donnée dans l'Acc.	2	1	-
MOV Rn, A	Copier l'Acc dans le registre	1	1	-
MOV Rn, direct	Copier la mémoire adressée dans le registre	2	2	-
MOV Rn, #data	Copier la donnée dans le registre	2	1	-
MOV direct, A	Copier l'Acc. dans la mémoire adressée	2	1	-
MOV direct, Rn	Copier le registre dans la mémoire adressée	2	2	-
MOV direct, direct	Copier mémoire dans mémoire	3	2	-
MOV direct, @Ri	Copier mémoire dans mémoire	2	2	-
MOV direct, #data	Copier donnée dans mémoire	3	2	-
MOV @Ri, A	Copier l'Acc. dans mémoire	1	1	-
MOV @Ri, direct	Copier mémoire dans mémoire	2	2	-
MOV @Ri, #data	Copier donnée dans mémoire	2	1	-
MOV DPTR, #data16	Copier la donnée 16bits dans DPTR	3	2	-
PUSH direct	Empiler la mémoire adressée	2	2	
POP direct	Dépiler un octet dans la mémoire adressé	2	2	
XCH A, Rn	Echanger l'Acc. avec le registre	1	1	
XCH A, direct	Echanger A avec la mémoire	2	1	
XCH A, @Ri	Echanger A avec la mémoire	1	1	
XCHD A, @Ri	Echange le digit bas de A avec celui de la mémoire	1	1	
MOVX A,@Ri	Copier la mémoire externe dans A	1	2	
MOVX A,@DPTR	Copier la mémoire externe adressée par DPTR dans A	1	2	
MOVX @Ri,A	Copier A dans la mémoire externe	1	2	
MOVX @DPTR,A	Copier A dans la mémoire externe pointée par DPTR	1	2	
MOVC A,@A+DPTR	Copier la mémoire externe pointée par A+DPTR dans A	1	2	
MOVC A,@A+PC	Copier la mémoire externe pointée par A+PC dans A	1	2	
Instruction Arithmétiques				
ADD A, Rn	A = A + Rn	1	1	C, AC, O
ADD A, direct	A = A + [direct]	2	1	C, AC, O
ADD A, @Ri	A = A + [Ri]	1	1	C, AC, O
ADD A, #data	A = A + donnée immédiate	2	1	C, AC, O
ADDC A, Rn	A = A + Rn + C	1	1	C, AC, O
ADDC A, direct	A = A + [direct] + C	2	1	C, AC, O
ADDC A, @Ri	A = A + [Ri] + C	1	1	C, AC, O
ADDC A, #data	A = A + donnée immédiate + C	2	1	C, AC, O
SUBB A, Rn	A = A – (Rn + C)	1	1	C, AC, O
SUBB A, direct	A = A - ([direct] + C)	2	1	C, AC, O
SUBB A, @Ri	A = A - ([Ri] + C)	1	1	C, AC, O
SUBB A, #data	A = A – (donnée immédiate + C)	2	1	C, AC, O
INC @Ri	Incrémenter la mémoire pointée par Ri	1	1	-
INC A	Incrémenter A	1	1	-
INC direct	Incrémenter mémoire adressée	2	1	-
INC Rn	Incrémenter registre	1	1	-
INC DPTR	Incrémenter le registre DPTR	1	2	-
DEC @Ri	Décrémenter la mémoire pointée par Ri	1	1	-
DEC A	Décrémenter A	1	1	-
DEC direct	Décrémenter mémoire adressée	2	1	-
DEC Rn	Décrémenter registre	1	1	-
MUL AB	A x B $\rightarrow$ B:A (non signé)	1	4	C, O
DIV AB	A / B $\rightarrow$ A , reste $\rightarrow$ B (non signé)	1	4	C, O
DA A	Ajustement décimal de A après une addition	1	1	C
Instructions logiques				
ANL A, #data	A = A ET donnée	2	1	-
ANL A, @Ri	A = A ET mémoire	1	1	-
ANL A, direct	A = A ET mémoire	2	1	-

ANL A, Rn	A = A ET registre	1	1	-
ANL direct, #data	Mémoire = mémoire ET donnée	3	2	-
ANL direct, A	Mémoire = mémoire ET A	2	1	-
ORL A, #data	A = A OU donnée	2	1	-
ORL A,@Ri	A = A OU mémoire	1	1	-
ORL A, direct	A = A OU mémoire	2	1	-
ORL A, Rn	A = A OU registre	1	1	-
ORL direct, #data	Mémoire = mémoire OU donnée	3	2	-
ORL direct, A	Mémoire = mémoire OU A	2	1	-
XRL A, #data	A = A XOR donnée	2	1	-
XRL A, @Ri	A = A XOR mémoire	1	1	-
XRL A, direct	A = A XOR mémoire	2	1	-
XRL A, Rn	A = A XOR registre	1	1	-
XRL direct, #data	Mémoire = mémoire XOR donnée	3	2	-
XRL direct, A	Mémoire = mémoire XOR A	2	1	-
CLR A	Mettre A à zéro	1	1	-
CPL A	Complémenter A	1	1	-
RL A	Rotation à gauche de A	1	1	-
RLC A	Rotation à gauche de A à travers le Cy	1	1	C
RR A	Rotation à droite de A	1	1	-
RRC A	Rotation à droite de A à travers Cy	1	1	C
SWAP A	Echange les digits de A	1	1	-
Instructions booléennes sur bits				
CLR bitadr	Clear bit	2	1	-
CLR C	Clear Cy	1	1	C=0
CPL bitadr	Complémenter bit	2	1	-
CPL C	Complémenter Cy	1	1	C
SETB bitadr	Positionner bit	2	1	-
SETB C	Positionner Cy	1	1	-
ANL C, /bitadr	C = C ET /bit	2	2	C
ANL C, bitadr	C = C ET bit	2	2	C
ORL C, /bit	C = C OR /bit	2	2	C
ORL C, bit	C = C OR bit	2	2	C
MOV bitAdr, C	Copier l'indicateur Cy dans le bit adressé	2	2	-
MOV C, bit Adr	Copier le bit adressé dans l'indicateur Cy	2	1	C
Instructions de branchement				
AJMP addr11	Saut avec adresse absolue sur 11 bits	2	2	
LJMP addr16	Branchement inconditionnel à adresse 16 bits	3	2	
SJMP rel	Branchement court avec adressage relatif	2	2	
JMP @A+DPTR	Saut à l'instruction d'adresse par A+DPTR	1	2	
JZ rel	Saut relatif si A= 0	2	2	
JNZ rel	Saut relatif si A ≠ 0	2	2	
JC rel	Saut relatif si Cy = 1	2	2	
JNC rel	Saut relatif si Cy = 0	2	2	
JB bitadr, rel	Saut relatif si bit = 1	3	2	
JNB bitadr, rel	Saut relatif si bit = 0	3	2	
JBC bitadr,rel	Saut relatif si bit = 1 puis RAZ le bit	3	2	
CJNE @Ri, #data, rel	Comparer mémoire et donnée et saut si différent	3	2	
CJNE A, #data, rel	Comparer A et donnée et saut si différent	3	2	
CJNE A, direct, rel	Comparer A et mémoire et saut si différent	3	2	
CJNE Rn, #data, rel	Comparer registre et donnée et saut si différent	3	2	
DJNZ direct, rel	Décrémenter mémoire et saut si différent de zéro	3	2	
DJNZ Rn, rel	Décrémenter registre et saut si différent de zéro	2	2	
Instruction d'appel et de retour de procédure				
ACALL adr11	Appel procédure avec adresse absolue sur 11 bits	2	2	
LCALL addr16	Appel de procédure adresse 16 bits	3	2	
RET	Retour de procédure	1	2	
IRET	Retour de procédure d'interruption	1	2	
Autre				
NOP	Ne fait rien	1	1	

V Les interruptions

Le 8051 possède 5 sources d'interruption, le 8052 en possède une de plus :

- Evénement externe sur l'entrée INT0 (P3.2)
- Evénement externe sur l'entrée INT1 (P3.3)
- Débordement du TIMER0
- Débordement du TIMER1
- Emission ou réception d'u caractère sur le port série
- Débordement du Timer 2 ou déclenchement du mode chargement/capture de ce dernier par l'entrée T2EX (P1.1)

V.1 Gestion des interruptions

D'une manière générale, chaque interruption est gérée par 3 bits situés dans des registres SFR :

- Un drapeau, qui est un bit positionné quant l'événement déclencheur intervient,
- Un bit de validation : Quand l'événement déclencheur intervient si le bit de validation est positionné, l'interruption peut avoir lieu sinon elle est ignorée,
- Un bit de priorité : Avec un bit, chaque interruption a 2 niveaux de priorité, (0=basse), (1=haute). Quand l'événement déclencheur intervient, même si l'interruption en question est validée, si une interruption de même ou de niveau supérieur est en train d'être exécutée, alors notre l'interruption devra attendre.

Le bit EA situé à la position 7 du registre SFR IE permet la validation/inhibition globale de toutes les interruptions.

V.2 Déroulement d'une interruption

Quand un événement déclencheur d'une interruption se produit, le drapeau correspondant est positionné (voir tableau ci-dessous), si le bit de validation est positionné et l'interruption est prioritaire alors l'interruption est déclenchée et les taches suivantes sont réalisées :

- La valeur du *Program Counter* est empilée

- Les interruptions de même ou de priorité inférieures sont masquées

- Dans le cas des interruptions INT0/1 ou TIMER0/1, le drapeau correspondant est remis à zéro. Pour les autre interruptions, le programmeur doit remettre lui même ce drapeau à zéro au début de la routine d'interruption.

- Le PC est chargée par l'adresse du vecteur d'interruption correspondant ce qui transfert l'exécution au début de ce vecteur ou doit résider la routine d'interruption.

- A l'exécution de l'instruction RETI qui doit terminer chaque routine d'interruption, le processeur dépile le PC pour retrouver sa place dans le programme principal et remet le statut des interruptions dans l'état où il l'a trouvé avant de commencer cette interruption

d'interruption	Flag	Validation	vecteur	Priorité	EDG
INT0 (P3.2)	IE0 (TCON.1)	EX0 (IE.0)	0003H	PX0 (IP.0)	IT0 (TCON.0)
INT1 (P3.3)	IE1 (TCON.3)	EX1 (IE.2)	0013H	PX1 (IP.2)	IT1 (TCON.2)
TIMER0 Overflow	TF0 (TCON.5)	ET0 (IE.1)	000BH	PT0 (IP.1)	
TIMER1 Overflow	TF1 (TCON.7)	ET1 (IE.3)	001BH	PT1 (IP.3)	
PORT SERIE	RI ou TI (SCON 0 et 1)	ES (IE.4)	0023H	PS (IP.4)	
TIMER2 Overflow	TF2 (T2CON.7)	ET2 (IE.5)	002BH	PT2 (IP.5)	
TIMER2 capture/reload	EXF2 (T2CON.6)	EXEN2(T2CON.3) ET2(IE.5)			↓ sur T2EX(P1.1)
Validation globale de toutes les interruptions		EA (IE.7)			

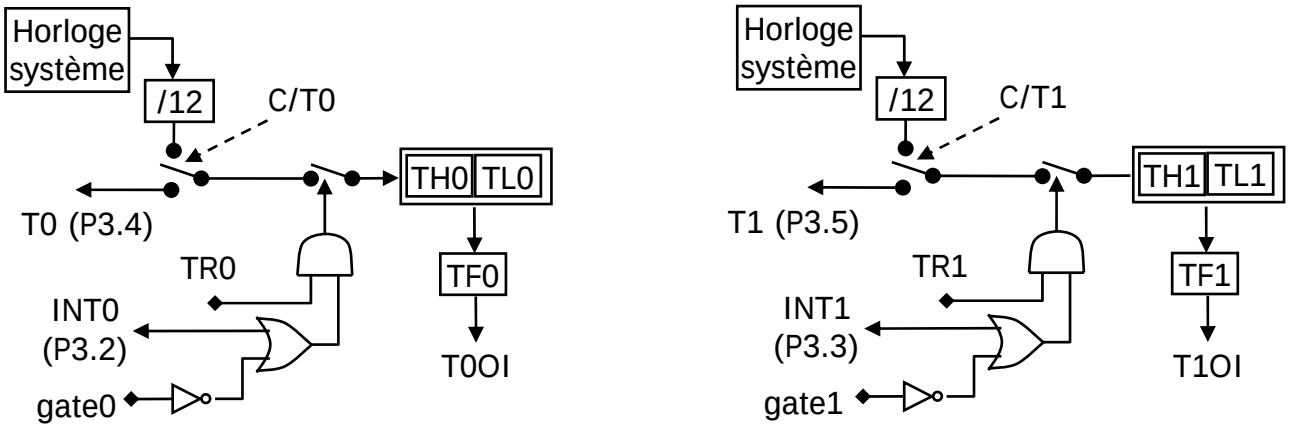
Tableau V.1 : Interruptions du 8051/8052

VI Les timers

Le 8051 possède deux timers 16 bits TIMER0 et TIMER1 dont le fonctionnement est déterminé par les registres de configuration TMOD et TCON. Les signaux horloges de ces circuits proviennent soit de l'horloge interne du système (mode Timer) soit des entrées externes T0 (P3.4) et T1 (P3.5) (mode Compteur). Le débordement de ces timers positionne les drapeaux TF0 ou TF1 qui déclenchent les interruptions correspondantes si elles ne sont pas masquées.

Les deux Timers ont un fonctionnement identique. il sont accessibles par l'intermédiaire des registres 8 bits TLx et THx qui constituent leurs parties basse et haute

SFR	**Description**	**Adresse**
TH0	Timer 0 High Byte	8Ch
TL0	Timer 0 Low Byte	8Ah
TH1	Timer 1 High Byte	8Dh
TL1	Timer 1 Low Byte	8Bh
TCON	Timer Control	88h
TMOD	Timer Mode	89h



TCON (88h)

Bit	Nom	Adr.	description
7	TF1	8Fh	Ce drapeau est positionné par le microcontrôleur quand le TIMER1 déborde
6	TR1	8Eh	Ce bit permet de démarrer/arrêter (1/0) le TIMER 1 (voir aussi le bit GATE1 de TMOD)
5	TF0	8Dh	Ce drapeau est positionné par le microcontrôleur quand le TIMER 0 déborde
4	TR0	8Ch	Ce bit permet de démarrer/arrêter (1/0) le TIMER 0 (voir aussi le bit GATE0 de TMOD)

TMOD (89h) SFR

Bit	Nom	description
7	GATE1	Quand ce bit est positionné, TIMER1 ne peut fonctionner que si l'entrée INT1 (P3.3) est haute. Quand il est à zéro, TIMER1 peut fonctionner indépendamment de l'entrée INT1
6	C/T1	Si bit est positionné, TIMER1 fonctionne en compteur d'événements sur l'entrée T1 (P3.5). Si ce bit est à zéro, TIMER1 est incrémenté par l'horloge système
5	T1M1	Timer1 mode (voir ci-dessous)
4	T1M0	Timer1 mode (voir ci-dessous)
3	GATE0	Quand ce bit est positionné, TIMER0 ne peut fonctionner que si l'entrée INT0 (P3.2) est haute. Quand il est à zéro, TIMER0 peut fonctionner indépendamment de l'entrée INT0
2	C/T0	Si bit est positionné, TIMER0 fonctionne en compteur d'événements sur l'entrée T0 (P3.4). Si ce bit est à zéro, TIMER0 est incrémenté par l'horloge système
1	T0M1	Timer0 mode (voir ci-dessous)
0	T0M0	Timer0 mode (voir ci-dessous)

TxM1	**TxM0**	Mode	Description
0	0	0	13-bit Timer.
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

Mode 0 : 13 bits

Ce mode sert surtout à garder la compatibilité avec le 8048, le prédécesseur du 8051. Il est peut utilisé dans les nouveaux développements. Seuls les bits 0 à 4 de TLx sont utilisés.

Mode 1 : 16 bits

C'est le mode le plus utilisé. Avec 16 bits, on a une période de comptage de 65536 cycles machine.

Mode 2 : 8 bits auto-reload

Dans ce mode c'est TLx qui fonctionne en Timer/compteur, Quand il déborde, au lieu d'être chargé par 0, il est chargé par la valeur contenue dans THx.
Ce mode est souvent utilisé pour définir la vitesse de communication du port série.

Mode 3 : Split-Timer

Dans ce mode Les deux registres TL0 et TH0 du Timer 0 fonctionnent comme deux timers 8 bits indépendants.
TL0 utilisent tous les bits de contrôle de TIMER0 et déclanche (quand il déborde) l'interruption *T0OI* : *TIMER0 Overflow interrupt*.
TH0 emprunte les bits TR1 et TF1 à TIMER1 et déclanche (quand il déborde) l'interruption *T1OI : TIMER1 Overflow interrupt* .
Placé en mode 3, TIMER1 est arrêté exactement comme si on a fait TR1 = 0. Par contre, Si TIMER0 est en mode 3, TIMER1 peut être utilisé en mode 0,1 ou 2 mais il ne peut être contrôlé par le bit TR1 car il est affectée à TH0 et il ne déclenche pas d'interruption quand il déborde.

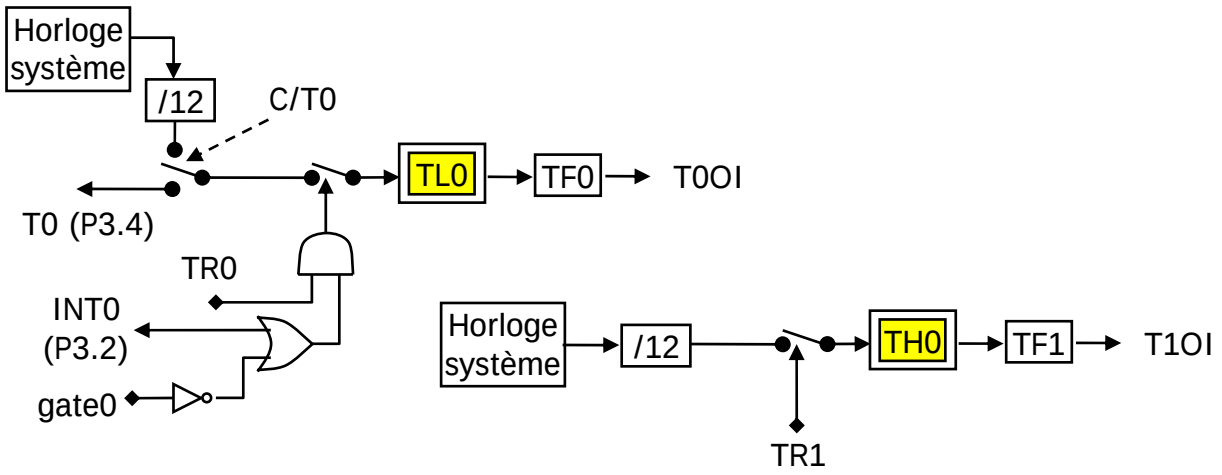


Figure VI.1 : TIMER0 en mode 3

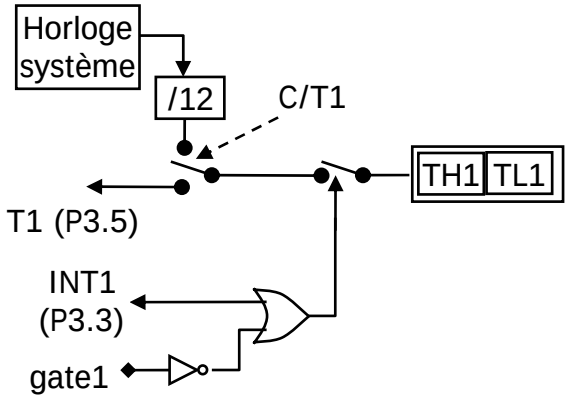


Figure VI.2 : TIMER1 quand TIMER0 est en mode 3

VI.1 Le TIMER2 du 8052

Le 8052 contient 3 timers; TIMER0, TIMER1 et TIMER2. Les deux premiers fonctionnent exactement comme ceux du 8051. TIMER2 est géré par les registres SFR suivants,

SFR	description	adr
T2CON	Contrôle de T2	C8h
TL2	poids faible du Timer 2	CDh
TH2	Poids fort du Timer 2	CEh
RCAP2L	Capture/recharge de TL2	CBh
RCAP2H	Capture/recharge du TH2	CCh

VI.1.1 TIMER2 en mode Auto-Reload

A la différence de TIMER0 et TIMER1, TIMER2 possède un mode de recharge (*auto-reload)* complètement 16 bits. Quand un auto-reload intervient, TL2 est chargé par le contenu du registre RCAP2L et TH2 est chargé par la valeur du registre RCAP2H.

Un auto-reload intervient soit après débordement soit après un front descendant sur T2EX (P1.1), voir tableau ci-dessus.

VI.1.2 TIMER2 en mode capture

Quand une capture intervient, le contenu de TIMER2 (TH2/TL2) est recopié dans la paire de registre RCAP2H/RCAP2L.

Une capture intervient au front descendant sur T2EX (P1.1), voir tableau ci-dessus.

T2CON

BIT	NOM	ADR	DESCRIPTION
7	TF2	CFh	Timer 2 Overflow. Ce bit est le drapeau de l'interruption T2OI *(TIMER2 Overflow Interrupt)*, il positionné quand TIMER2 déborde, si l'interruption est validée par ET2 (IE.5), elle sera déclenchée. Quand TIMER2 est utilisé comme générateur de rythme pour le port série, TF2 n'est pas positionné au débordement.
6	EXF2	CEh	Timer 2 External Flag. C'est le drapeau extérieur de l'interruption T2CRI *(TIMER2 Capture Reload Interrupt)*. Si le bit EXEN2 = 1, EXF2 sera positionné par un front descendant sur l'entrée T2EX (P1.1) qui provoque une recharge ou une capture de TIMER2. Si l'interruption est validée par ET2 (IE.5), elle sera déclenchée. T2OI et T2CRI on le même vecteur d'interruption. (Voir Tableau V.1 : page22)
5	RCLK	CDh	Timer 2 Receive Clock. Quand ce bit est positionné, TIMER2 est utilisé pour déterminer le rythme de réception du port série. S'il n'est pas positionné, c'est TIMER1 qui joue ce rôle.
4	TCLK	CCh	Timer 2 Transmit Clock. Quand ce bit est positionné, TIMER2 est utilisé pour déterminer le rythme de transmission du port série. S'il n'est pas positionné, c'est TIMER1 qui joue ce rôle.
3	EXEN2	CBh	Timer 2 External Enable. Si ce bit est positionné, un front descendant sur T2EX (P1.1) déclenche une recharge ou une capture de TIMER2. Le drapeau EXF2 est levé et l'interruption T2CRI sera déclenchée si elle est validée.
2	TR2	CAh	Timer 2 Run. Ce bit de démarrer ou d'arrêter TIMER2.
1	C/T2	C9h	Timer 2 Counter/Interval Timer. Si bit est positionné, TIMER2 fonctionne en compteur d'événements (front descendant) sur l'entrée T2 (P1.0). Si ce bit est à zéro, TIMER2 fonctionne en mode TIMER, il est incrémenté par l'horloge système
0	CP/RL2	C8h	Timer 2 Capture/Reload. Si ce bit est à zéro, un *auto-reload* intervient au débordement de TIMER2 ou au front descendant sur T2EX (P1.1). Si ce bit est positionné il y a capture de TIMER2 au front descendant sur T2EX (P1.1)

Tableau VI.1 : T2CON ; registre de control de TIMER2

VI.1.3 Les drapeau de l'interruption TIMER2

Les deux interruptions T2OI et T2CRI de TIMER2 ont chacune son drapeau mais elle ont le même vecteur d'interruption. Pour ce fait Quand une interruption intervient, le processeur se branche au vecteur d'interruption mais ne touche pas aux drapeaux afin que le programmeur puisse déterminer de quelle interruption il s'agit. Il revient alors au programme utilisateur de baisser le drapeau avant de quitter la routine d'interruption pour éviter que l'interruption ne soit déclenchée de nouveau, en effet, durant l'exécution de l'instruction "RETI", le processeur revalide les interruptions de même priorité.

VII Le port série

Le 8051 dispose d'un port série qui permet de communiquer avec l'exterieur sur les bornes RxD (P3.0) et TxD (P3.1).

La configuration du port se fait par le registre SCON et la lecture écriture dans le port se fait par le registre SBUF.

Le registre SBUF est une passerelle vers deux registres physiquement séparés, un pour la réception et l'autre pour la transmission.

A la réception, le port possède un *buffer* de 1 octet, il en résulte que le port peut commencer à recevoir un nouvel octet avant que l'octet présent dans le registre de réception ne soit lu. Néanmoins, à la fin de la réception du nouvel octet, celui-ci est copié dans le registre de réception en écrasant l'ancien octet.

Bit	Name	Adr	Description
7	SM0	9Fh	Détermine le mode de fonctionnement (voir tableau ci-dessous)
6	SM1	9Eh	Détermine le mode de fonctionnement (voir tableau ci-dessous)
5	SM2	9Dh	Validation du mode multiprocesseur
4	REN	9Ch	Validation de réception
3	TB8	9Bh	$9^{ème}$ bit à transmettre en mode UART 9 bits
2	RB8	9Ah	$9^{ème}$ bit reçu en mode UART 9 bits
1	TI	99h	Drapeau de fin de transmission
0	RI	98h	Drapeau de fin de réception

Tableau VII.1 : SCON : registre de configuration du port série

SM0	SM1	Mode	Explanation	Baud Rate
0	0	0	Registre à décalage 8 bits	Oscillator / 12
0	1	1	8-bit UART	Set by Timer 1 (*)
1	0	2	9-bit UART	Oscillator / 32 (*)
1	1	3	9-bit UART	Set by Timer 1 (*)

Tableau VII.2 : mode de fonctionnement --- (*) vitesse double si SMOD = 1

VII.1 Modes de fonctionnement

Le port série peut fonctionner selon 4 modes différents. Le choix des modes se fait par les bits SM0 et SM1 du registre SCON :

Mode 0 : 8 bit Shift Register

8 bits sont transmis sur TxD ou reçus sur RxD. La vitesse de transmission est Fosc/12

Mode 1 : 8 bit UART

10 bits sont transmis sur TxD ou reçus sur RxD :
- Un bit de start (toujours 0)
- 8 bits de donnée (LSB d'abord)
- Un bit de stop (toujours 1)

En réception le bit de stop copié à la position RB8 du registre SCON
Pour le 8051, la vitesse de transmission est fixée par TIMER1. Pour le 8052 on peut utiliser soit TIMER1 soit TIMER2. Attention, le bit SMOD (TCON.7) double la vitesse.

Mode 2 : 9 bit UART

11 bits sont transmis sur TxD ou reçus sur RxD :
- Un bit de start (toujours 0)
- 8 bits de donnée (LSB d'abord)
- Un bit programmable appelé $9^{ème}$ bit
- Un bit de stop (toujours 1)

En transmission, le bit TB8 de SCON est transmis comme $9^{ème}$ bit,
En réception, le $9^{ème}$ bit est recopié dans le bit RB8 de SCON. Le stop bit est ignoré.
La vitesse de transmission est fosc/64 ou fosc/32 selon la valeur du bit SMOD

Mode 3: 9 bit UART

Le mode 3 est identique au mode 2 à part le fait que la vitesse de transmission est fixée

par TIMER1 ou TIMER2

Les modes 2 et 3 sont utilisés essentiellement en mode multiprocesseur où le 9ème bit est utilisé pour adresser un processeur parmi 2 lors d'une communication multipoints.

VII.2 Transmission d'un Octet

Dans tous les modes, une transmission commence après chaque écriture dans le registre SBUF. La transmission se fait en série, elle n'est donc pas instantanée, le processeur nous informe de la fin de transmission en positionnant le flag TI. Ce bit peut déclencher l'interruption du port série si son bit de validation ES (IE.4) est positionné, il ne faut pas oublier de le remettre à zéro avant de commencer une nouvelle transmission.

VII.3 Réception d'un octet

Pour pouvoir recevoir des données, il faut que le bit de validation de réception REN soit positionné.

- En mode 0, une réception est initiée par la mise à 1 du flag RI
- Dans les autres mode, la réception démarre à l'arrivé du start bit sur l'entre RxD
- Dans tous les modes, la réception se termine par la mise à un du drapeau RI

VII.4 Définition de la vitesse de communication par Timer

Pour le 8051, la vitesse de communication est définie par le rythme de débordement de TIMER1. Pour le 8052, on peut utiliser soit TIMER1 soit TIMER2 soit les deux, un pour la réception l'autre pour la transmission.

Utilisation de TIMER1 :

Dans le cas de TIMER1, la vitesse de communication est définie par :

$$VC = \frac{2^{SMOD}}{32} \times \textit{fréquence de débordement}$$

D'une manière générale, TIMER1 est utilisé en mode Auto-Reload, dans ce cas, la vitesse de communication est définie par :

$$VC = \frac{2^{SMOD}}{32} \times \frac{fosc}{12 \times (256 - TH1)}$$

Vitesse de communication	Fréquence de l'oscillateur	SMOD	TH1
19200	11.059 MHz	1	FDh
9600	11.059 MHz	0	FDh
4800	11.059 MHz	0	FAh
2400	11.059 MHz	0	F4h
1200	11.059 MHz	0	E8h

Le tableau ci-contre donne la valeur de TH1 pour les vitesses de communication couramment utilisées et ceci pour un quartz de 10.059 MHz

Utilisation de TIMER2 :

La configuration de TIMER2 en générateur de rythme est réalisée à l'aide des bits RCLK et TCLK du registre T2CON. La vitesse de communication est donnée par :

$$VC = \frac{\textit{fréquence de débordement}}{16}$$

D'une manière générale, TIMER2 est utilisé en mode Auto-Reload, dans ce cas, la vitesse de transmission est définie par :

$$VC = \frac{fosc}{32 \times (65536 - RCAP2)}$$