MEGAWIN
MAKE YOU WIN

8051 内核-单片机

MPC89LE51_52_53_54_58_515 规格书

版本: A1.02

2015/09 version A1.02

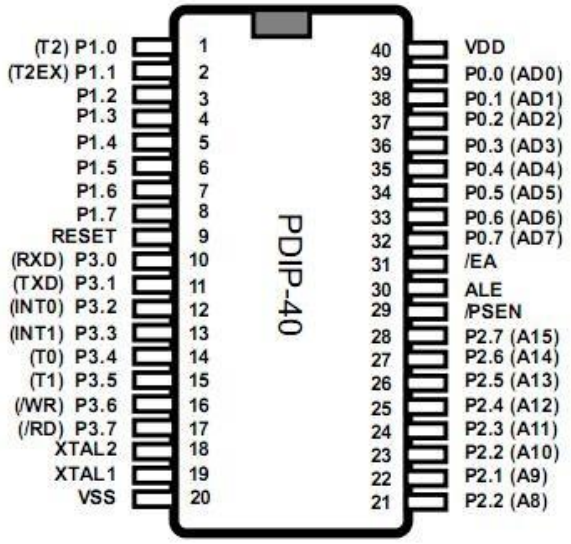
特性

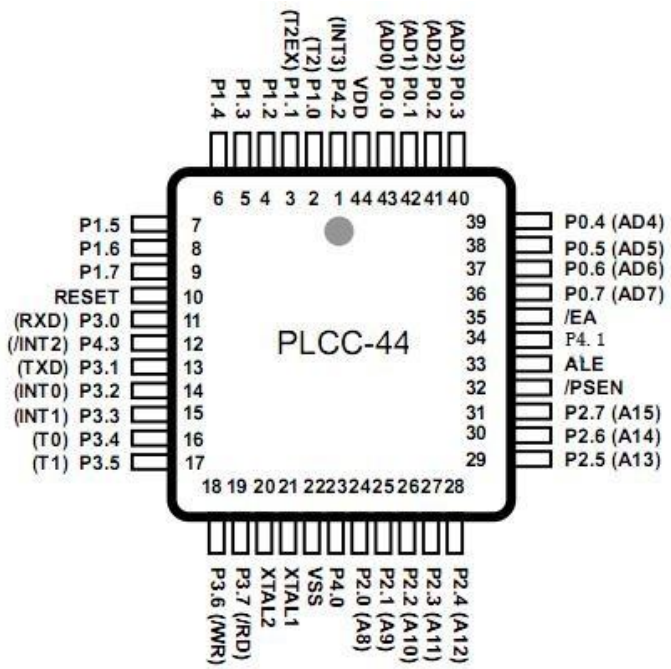
- 80C51 内核
- 可选 12T 或 6T 模式
- 最大工作频率位 48MHz@12T 或 24MHz@6T
- 程序空间: MPC89E/L51(**4KB**),MPC89E/L52(**8KB**),MPC89E/L53(**15KB**),MPC89E/L54(**16KB**)
  MPC89E/L58(**32KB**),MPC89E/L515(**63KB**)
- ISP 空间大小；可选 1K/2K/4K ISP 空间，对于 MPC89x51/52/53/54/58 是与数据空间共享，而 MPC89x53/515 是与应用程序空间共享
- IAP 空间大小； MPC89x51: 最大到 **10KB**,与 ISP 空间共享数据空间
  MPC89x52: 最大到 **6KB**,与 ISP 空间共享数据空间
  MPC89x53: 无
  MPC89x54: 最大到 **46KB**,与 ISP 空间共享数据空间
  MPC89x58: 最大到 **30KB**,与 ISP 空间共享数据空间
  MPC89x515: 无
  注: 要使用 IAP，必须设置 ISP 空间最小为 1K。
- 内嵌外部寻址 RAM(XDATA), MPC89x51/52/53(**256Byte**),MPC89x54/58/515(**1024Byte**);
- 两级代码加密保护
- 三个 16 位定时/计数器, Timer2 是一个向上/向下计数器，可编程时钟输出在 P1.0 口上
- 8 个中断源，4 级优先级
- 一组增强型 UART，提供帧错误检测和硬件地址识别
- 双 DPTR
- 15 位看门狗，8 位预分频。使能后，不能关闭
- 能耗控制；IDLE 模式和掉电模式；掉电模式能被 P3.2/P3.3/P4.2/P4.3 唤醒
- 低 EMI；可关闭 ALE 输出
- 4 组 8 位双向 I/O 口;对于 PLCC-44 和 PQFP-44 封装还有一组 4 位双向 I/O 口(P4)
- 芯片内程序/数据 FLASH 存储器
  - ✓ 超过 20,000 次擦写
  - ✓ 室温下数据保存大于 100 年
- 工作电压:
  MPC89E51/52/53/54/58/515: 4.5V~5.5V
  MPC89L51/52/53/54/58/515: 2.4V~3.6V，在对 FLASH 写操作(ISP)时，最小电压为 2.7V。
- 内建低压复位电路
- 最大工作频率
  可选 12T 或 6T 模式
  工业级，最大到 48MHz@12T, 24MHz@6T
- 封装:
  PDIP-40: MPC89x5xAE
  PLCC-44: MPC89x5xAP
  LQFP-44: MPC89x5xAD44

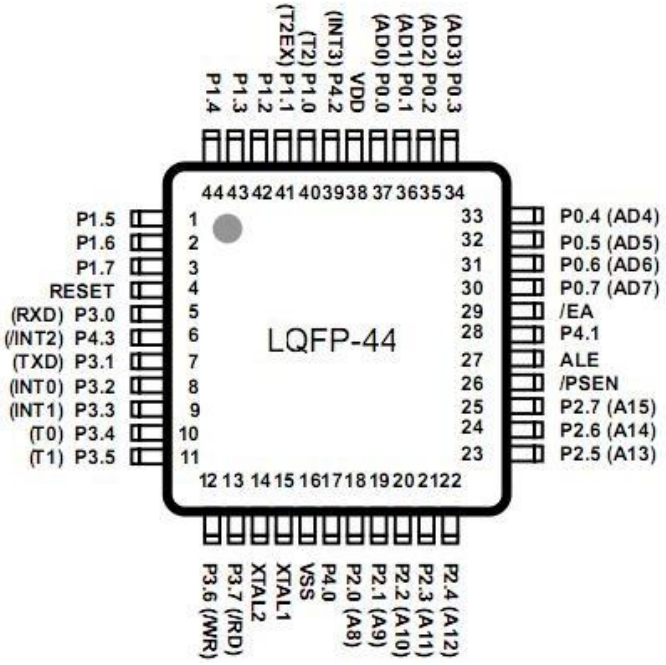
目录

1 引脚

1.1 引脚图


PDIP-40
(T2) P1.0 1
(T2EX) P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RESET 9
(RXD) P3.0 10
(TXD) P3.1 11
(INT0) P3.2 12
(INT1) P3.3 13
(T0) P3.4 14
(T1) P3.5 15
(/WR) P3.6 16
(/RD) P3.7 17
XTAL2 18
XTAL1 19
VSS 20
40 VDD
39 P0.0 (AD0)
38 P0.1 (AD1)
37 P0.2 (AD2)
36 P0.3 (AD3)
35 P0.4 (AD4)
34 P0.5 (AD5)
33 P0.6 (AD6)
32 P0.7 (AD7)
31 /EA
30 ALE
29 /PSEN
28 P2.7 (A15)
27 P2.6 (A14)
26 P2.5 (A13)
25 P2.4 (A12)
24 P2.3 (A11)
23 P2.2 (A10)
22 P2.1 (A9)
21 P2.2 (A8)


PLCC-44
P1.4 P1.3 P1.2 (T2EX) P1.1 (T2) P1.0 (INT3) P4.2 VDD (AD0) P0.0 (AD1) P0.1 (AD2) P0.2 (AD3) P0.3
6 5 4 3 2 1 44 43 42 41 40
P1.5 7
P1.6 8
P1.7 9
RESET 10
(RXD) P3.0 11
(/INT2) P4.3 12
(TXD) P3.1 13
(INT0) P3.2 14
(INT1) P3.3 15
(T0) P3.4 16
(T1) P3.5 17
39 P0.4 (AD4)
38 P0.5 (AD5)
37 P0.6 (AD6)
36 P0.7 (AD7)
35 /EA
34 P4.1
33 ALE
32 /PSEN
31 P2.7 (A15)
30 P2.6 (A14)
29 P2.5 (A13)
18 19 20 21 22 23 24 25 26 27 28
P3.6 (/WR) P3.7 (/RD) XTAL2 XTAL1 VSS P4.0 P2.0 (A8) P2.1 (A9) P2.2 (A10) P2.3 (A11) P2.4 (A12)

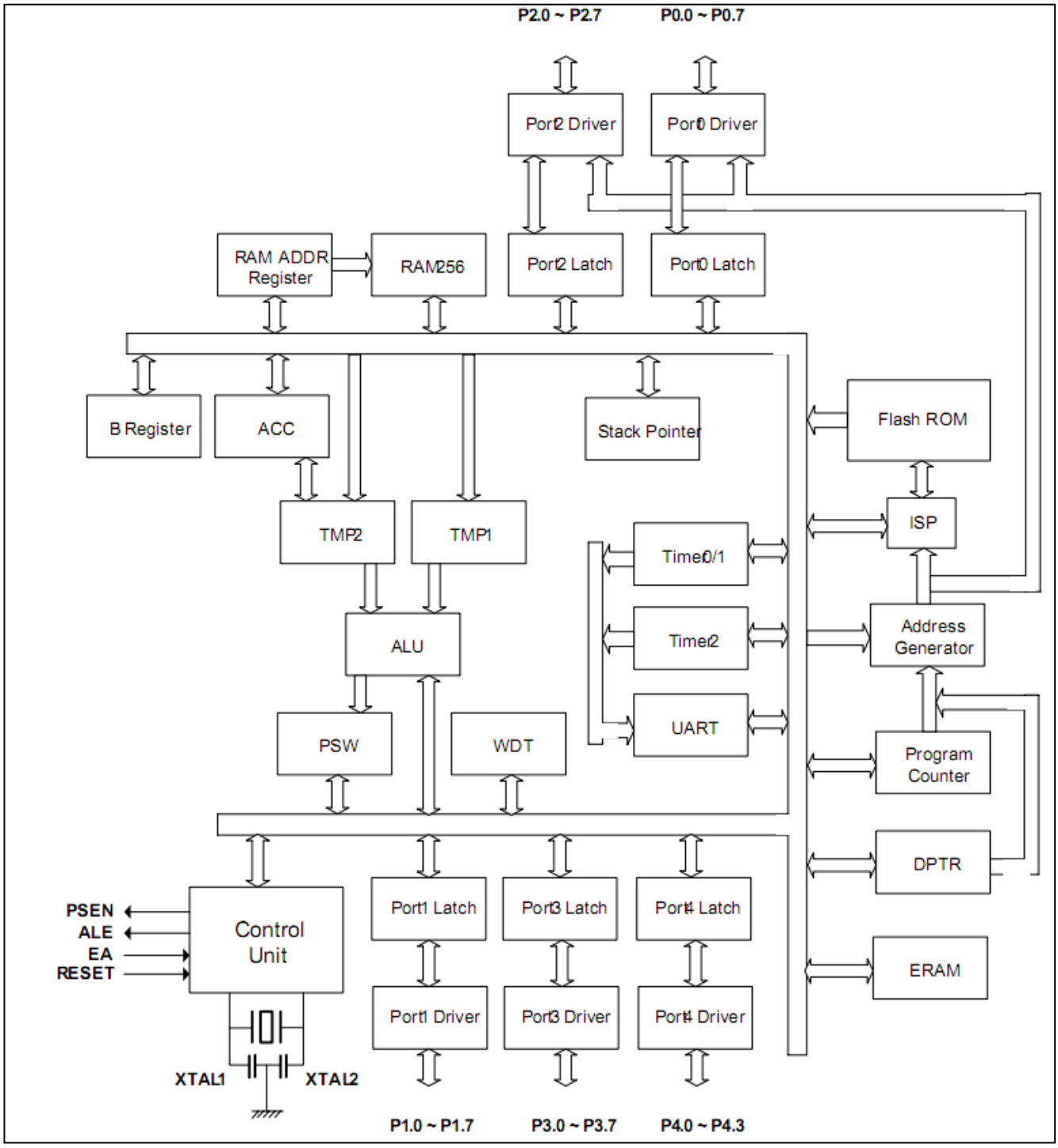

LQFP-44
P1.4 P1.3 P1.2 (T2EX) P1.1 (T2) P1.0 (INT3) P4.2 VDD (AD0) P0.0 (AD1) P0.1 (AD2) P0.2 (AD3) P0.3
44 43 42 41 40 39 38 37 36 35 34
P1.5 1
P1.6 2
P1.7 3
RESET 4
(RXD) P3.0 5
(/INT2) P4.3 6
(TXD) P3.1 7
(INT0) P3.2 8
(INT1) P3.3 9
(T0) P3.4 10
(T1) P3.5 11
33 P0.4 (AD4)
32 P0.5 (AD5)
31 P0.6 (AD6)
30 P0.7 (AD7)
29 /EA
28 P4.1
27 ALE
26 /PSEN
25 P2.7 (A15)
24 P2.6 (A14)
23 P2.5 (A13)
12 13 14 15 16 17 18 19 20 21 22
P3.6 (/WR) P3.7 (/RD) XTAL2 XTAL1 VSS P4.0 P2.0 (A8) P2.1 (A9) P2.2 (A10) P2.3 (A11) P2.4 (A12)

1.1 引脚定义

引脚名	引脚号			类型	描述
	DIP-40	PLCC-44	LQFP-44		
P0.0(AD0) P0.1(AD1) P0.2(AD2) P0.3(AD3) P0.4(AD4) P0.5(AD5) P0.6(AD6) P0.7(AD7)	39 38 37 36 35 34 33 32	43 42 41 40 39 38 37 36	37 36 35 34 33 32 31 30	I/O	Port0 是开集，双向 I/O 口。当向 Port0 写入 1 后，Port0 成为一高阻输入口。在寻址外部程序或数据时，Port0 复用为低 8 位地址或数据总线。
P1.0(T2) P1.1(T2EX) P1.2 P1.3 P1.4 P1.5 P1.6 P1.7	1 2 3 4 5 6 7 8	2 3 4 5 6 7 8 9	40 41 42 43 44 1 2 3	I/O	Port1 是带上拉电阻的双向 I/O 口。当向 Port1 写入 1 后，Port1 被内部上拉成高电平,可用在输入口。 P1.0(T2): 可用作 Timer2 的外部计数或时钟输出。 P1.1(T2EX):可用作 Timer2 的中断控制源
P2.0(A8) P2.1(A9) P2.2(A10) P2.3(A11) P2.4(A12) P2.5(A13) P2.6(A14) P2.7(A15)	21 22 23 24 25 26 27 28	24 25 26 27 28 29 30 31	18 19 20 21 22 23 24 25	I/O	Port1 是带上拉电阻的双向 I/O 口。当向 Port1 写入 1 后，Port1 被内部上拉成高电平,可用在输入口。在访问外部程序存储器和外部数据时分别作为地址高八位字节
P3.0(RXD) P3.1(TXD) P3.2(INT0) P3.3(INT1) P3.4(T0) P3.5(T1) P3.6(/WR) P3.7(/RD)	10 11 12 13 14 15 16 17	11 12 13 14 15 16 17 18	5 7 8 9 10 11 12 13	I/O	P3 是带内部上拉的双向 I/O 口,向 P3 口写入 1 时 P3 口被内部上拉为高电平,可用作输入口 P3 口还具有以下特殊功能 RxD(P3.0) 串行输入口 TxD(P3.1) 串行输出口 INT0(P3.2) 外部中断 0 INT1(P3.3) 外部中断 1 T0(P3.4) 定时器 0 外部输入 T1(P3.5) 定时器 1 外部输入 /WR(P3.6) 外部数据存储器写信号 /RD(P3.7) 外部数据存储器读信号

P4.0 P4.1 P4.2(/INT3) P4.3(/INT2)	无	23 34 1 12	17 28 39 6	I/O	P4 仅在 PLCC-44 和 PQFP-44 封装中有。 它是带上拉电阻的双向 I/O 口。 P4.2(/INT3)　外部中断 0 输入 P4.3(/INT2)　外部中断 1 输入
RESET	9	10	4	I	当该脚输入高电平超过 2 个机器周期时，芯片就会产生复位，內置下拉電阻
ALE	30	33	27	O	在访问外部存储器时 输出脉冲锁存地址的低字节
/PSEN	29	32	26	O	程序存储使能 当执行外部程序存储器代码时 PSEN 每个机器周期被激活两次 在访问外部数据存储器时 PSEN 无效 访问内部程序存储器时 PSEN 无
/EA	31	35	29	I	外部寻址使能: 在访问整个外部程序存储器时 EA 必须外部置低 。如果 EA 为高时 将执行内部程序除非程序计数器包含大于片内 FLASH 的地址。
XTAL1	19	21	15	I	晶振输入
XTAL2	18	20	14	O	晶振输出
VDD	40	44	38	P	电源
VSS	20	22	16	G	地

2 方框图



MPC89x5x 结构方框图

3 特殊功能寄存器

	8	9	A	B	C	D	E	F
F8								
F0	B							
E8								
E0	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8								
D0	PSW							
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2		
C0	XICON							*
B8	IP	SADEN						
B0	P3							
A8	IE	SADDR						IPH
A0	P2		AUXR1					
98	SCON	SBUF						
90	P1							
88	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	
80	P0	SP	DPL	DPH				PCON
	0	1	2	3	4	5	6	7

标号	地址	描述	7	6	5	4	3	2	1	0	初始值
P0	80H	I/O 口 0									11111111B
SP	81H	堆栈指针									00000111B
DPL	82H	数据指针低 8 位									00000000B
DPH	83H	数据指针高 8 位									00000000B
PCON	87H	电源控制	SMOD	SMOD0	--	POF	GF1	GF0	PD	IDL	00110000B
TCON	88H	Timer 控制	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000B
TMOD	89H	Timer 模式	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000B
TL0	8AH	T0 低 8 位									00000000B
TL1	8BH	T1 低 8 位									00000000B
TH0	8CH	T0 高 8 位									00000000B
TH1	8DH	T1 高 8 位									00000000B
AUXR	8EH	辅助寄存器							ERAM	AO	xxxxxx00B
P1	90H	I/O 口 1							T2EX	T2	11111111B
SCON	98H	串口控制	SM0/FE	M1	SM2	REN	TB8	RB8	TI	RI	00000000B
SBUF	99H	串口缓存									xxxxxxxxB
P2	A0H	I/O 口 2									11111111B
AUXR1	A2H	辅助寄存器 1					GF2			DPS	xxxx0xx0B
IE	A8H	中断使能	EA		ET2	ES	ET1	EX1	ET0	EX0	00000000B
SADDR	A9H	从地址									00000000B
P3	B0H	I/O 口 3	RD	WR	T1	T0	INT1	INT0	TXD	RXD	11111111B
IPH	B7H	中断优先级高位	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000B
IP	B8H	中断优先级高位			PT2	PS	PT1	PX1	PT0	PX0	xx000000B

SADEN	B9H	从地址屏蔽									00000000B
XICON	C0H	扩展中断控制	PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2	00000000B
T2CON	C8H	Timer2 控制	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL	00000000B
T2MOD	C9H	Timer2 模式							T2OE	DCEN	xxxxxx00B
RCAP2L	CAH	T2 捕获低字节									00000000B
RCAP2H	CBH	T2 捕获高字节									00000000B
TL2	CDH	T2 低字节									00000000B
TH2	CEH	T2 高字节									00000000B
PSW	D0H	程序状态字	CY	AC	F0	RS1	RS0	OV	-	P	00000000B
ACC	E0H	累加器									00000000B
WDTCR	E1H	看门狗控制	-	-	ENW	CLW	WIDL	PS2	PS1	PS0	xx000000B
IFD	E2H	ISP 数据									11111111B
IFADRH	E3H	ISP 地址高位									00000000B
IFADRL	E4H	ISP 地址低位									00000000B
IFMT	E5H	ISP 模式	--	-	-	-	-	MS2	MS1	MS0	xxxxx000B
SCMD	E6H	ISP 命令									xxxxxxxxB
ISPCR	E7H	ISP 控制	ISPEN	BS	SRST	-	-	ICK2	ICK1	ICK0	000xx000B
P4	E8H	I/O 口 4					EBH	EAH	E9H	E8H	xxxx1111B
B	F0H	B 寄存器									00000000B

辅助寄存器 AUXR 和 AUXR1

AUXR(8EH)　　初始置:xxxxxx00B

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
-	-	-	-	-	-	ERAM	AO

ERAM:　　定义是否隐藏内部扩展 RAM,而指向外部 RAM

0: 内部扩展 RAM 使能

1: 内部扩展 RAM 禁止。MOVX 指令总是指向外部 RAM

AO:　　0:　ALE 发出 OSC/6(12T),OSC/3(6T)固定频率的波形

1：　ALE 只在 MOVC 和 MOVX 指令指向外部存储空间的时候有效。

AUXR1(A2H)　　初始置:xxxx0xx0B

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
-	-	-	-	GF2	-	-	DPS

GF2:　　通用标志位

DPS:　　数据指针切换位

0: 数据指针 0 有效

1: 数据指针 1 有效

4 存储器

4.1 RAM

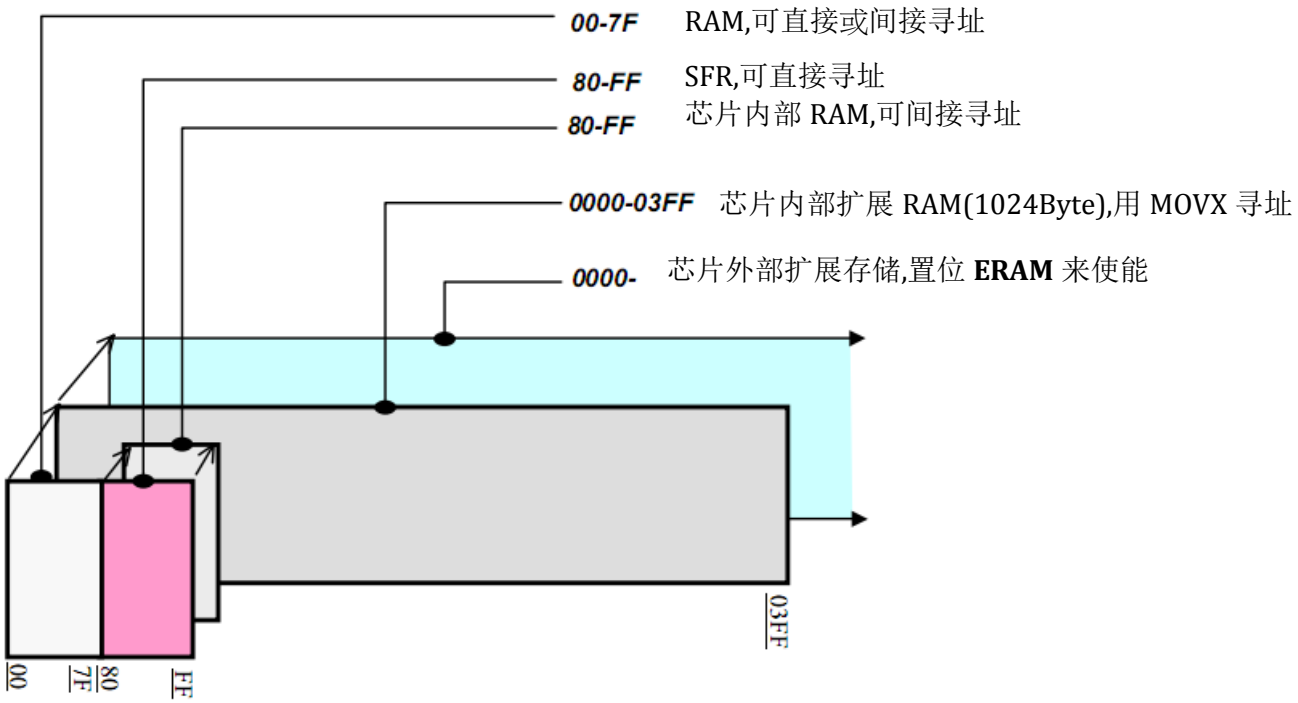
对于 MPC89x51/52/53，内建 512 字节 RAM. 对于 MPC89x54/58/515,内建 1280 字节 RAM.

用户可直接或间接寻址开始的 128 字节 RAM, 我们叫它直接 RAM,它的地址空间是 00h~7Fh.

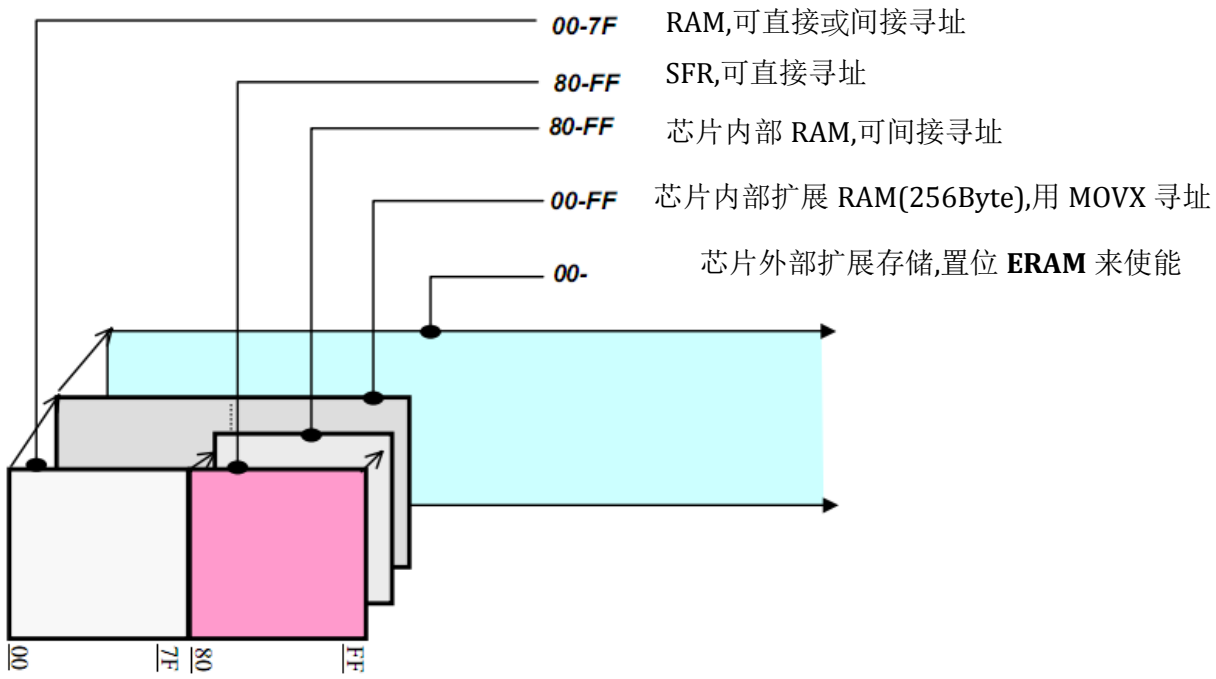
接下来的 128 字节 RAM,用户可以间接寻址到它。我们叫它间接 RAM,它的空间地址是 80h~FFh

其它的 RAM(**MPC89x51/52/53 是 256 字节，MPC89x54/58/515 是 1024 字节**)被叫做**扩展 RAM**,它占用的空间地址分别是 **00h~FFh(MPC89x51/52/53),000h~3FFh(MPC89x54/58/515)**,用户可以通过寄存器 **Ri** 或数据指针 **DPTR,**使用 *MOVX* 指令来访问它,如:**MOVX A,@R1 or MOVX A,@DPTR**。为了使 *MOVX* 指令能像一般的访问外部空间，用户可以将特殊寄存器 **AUXR 的 ERAM 位置为 1，**则 MOVX 就是访问外部空间，而不是访问扩展 RAM 了。

MPC89x54/58/515 RAM 空间



MPC89x51/52/53 RAM 空间



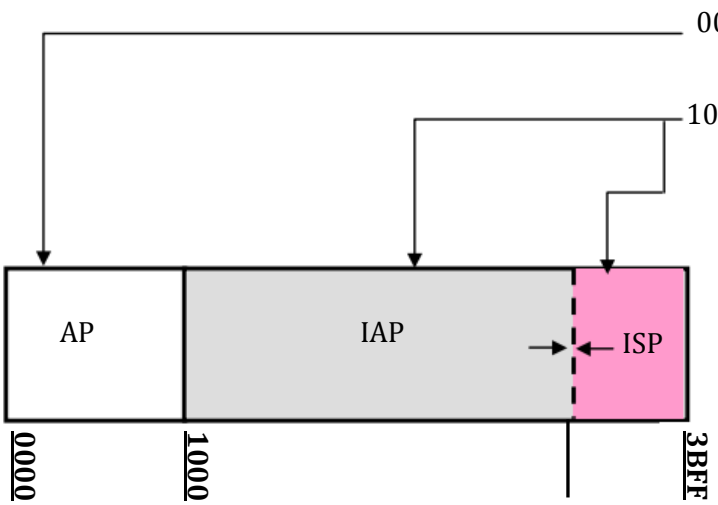
4.2 FLASH

对于 MPC89x51/52/53，总共有 15K 字节 FLASH,对于 MPC89x54/58/515，总共有 63K 字节 FLASH.

对于 MPC89x51，开始的 4K 是程序空间，紧接着的 11K 空间是 IAP 空间和 ISP 空间共享
对于 MPC89x52，开始的 8K 是程序空间，紧接着的 7K 空间是 IAP 空间和 ISP 空间共享
对于 MPC89x53，15K 空间是 AP 空间和 ISP 空间共享
对于 MPC89x54，开始的 16K 是程序空间，紧接着的 47K 空间是 IAP 空间和 ISP 空间共享
对于 MPC89x58，开始的 32K 是程序空间，紧接着的 21K 空间是 IAP 空间和 ISP 空间共享
对于 MPC89x515，63K 空间是 AP 空间和 ISP 空间共享

当程序计数器跨过 0FFFh(MPC89x51),1FFFh(MPC89x52),2BFFh/33FFh/37FFh/3BFFh(MPC89x53), 3FFFh(MPC89x54),7FFFh(MPC89x58),EBFFh/F3FFh/F7FFh/FBFFh(MPC89x515)，芯片将立刻从外部程序空间获取程序代码，而不考虑/EA 引脚的状态。这样，它就不再从内部 FLASH 里获取程序代码。
用户可以开发自己的 ISP 程序，并且将它写入 ISP 空间，该空间在 FLASH 的起始地址可通过配置**硬件選項**来设置。

MPC89x51 FLASH 空间

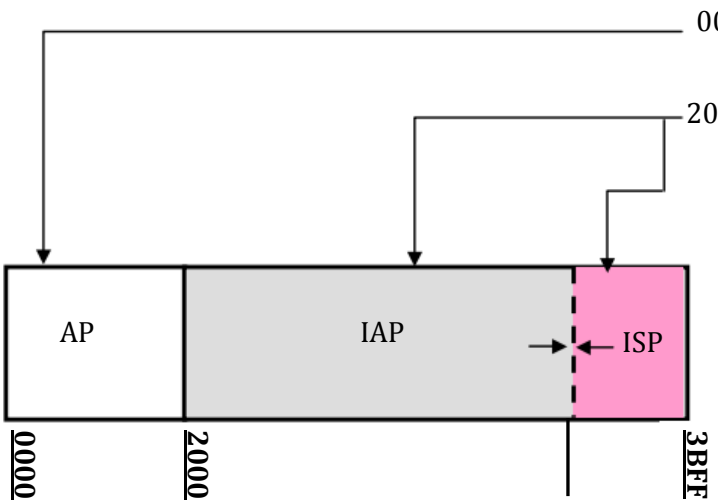


非易失性空间(IAP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

ISP 空间	IAP 空间
无	无
1KB	**10KB**
2KB	**9KB**
4KB	**7KB**

MPC89x52 FLASH 空间

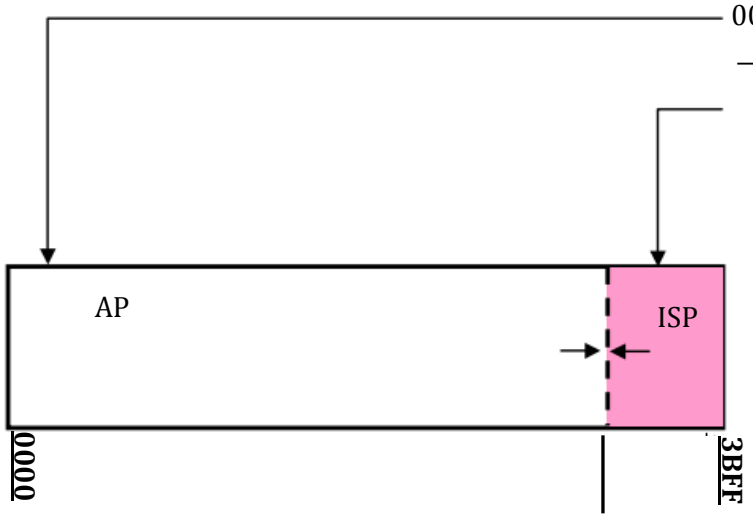


非易失性空间(IAP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

ISP 空间	IAP 空间
无	无
1KB	**10KB**
2KB	**9KB**
4KB	**7KB**

MPC89x53 FLASH 空间



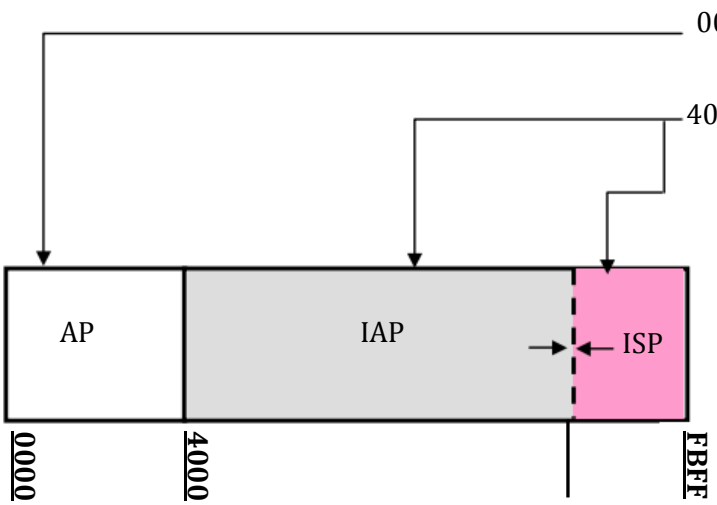
程序空间(AP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

:

ISP 空间	AP 空间
无	**15KB**
1KB	**14KB**
2KB	**13KB**

MPC89x54 FLASH 空间

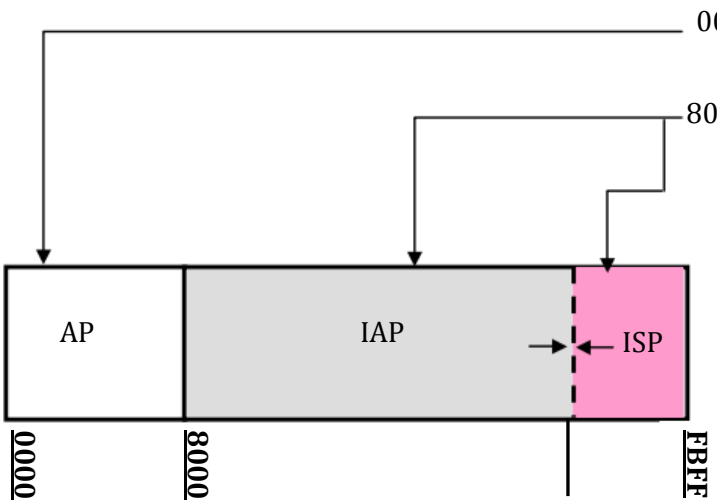


非易失性空间(IAP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

ISP 空间	IAP 空间
无	**无**
1KB	**46KB**
2KB	**45KB**
4KB	**43KB**

MPC89x58 FLASH 空间

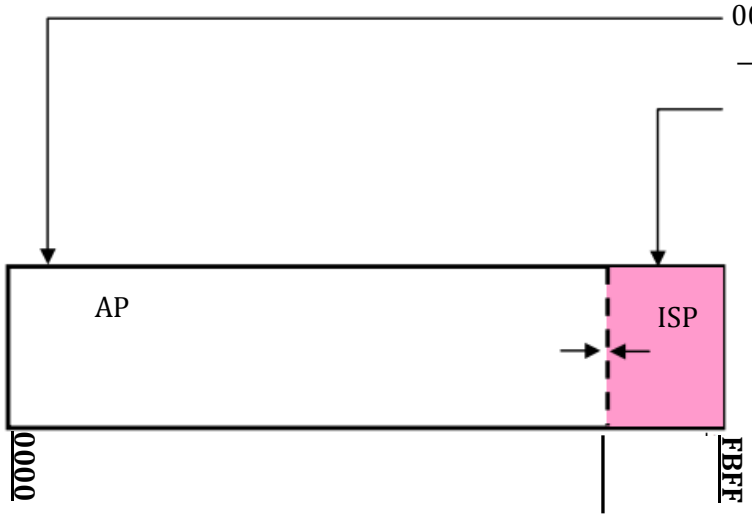


非易失性空间(IAP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

ISP 空间	IAP 空间
无	**无**
1KB	**30KB**
2KB	**29KB**
4KB	**27KB**

MPC89x515 FLASH 空间



程序空间(AP)和 ISP 空间共享.

ISP,IAP 空间可通过**硬件选项**配置为:

ISP 空间	AP 空间
无	**63KB**
1KB	**62KB**
2KB	**61KB**
4KB	**59KB**

4.3 硬件选项寄存器

通常使用通用编程器来写这两个寄存器，例如Hi-Lo系列的All-11,Leaper-48和笙泉提供的MCU编程器。

ISP 空间的起始地址

4K: ISP 空间是 **0x2C00~0x3BFF(MPC89x51/52/53)**,**0xEC00~0xFBFF(MPC89x54/58/515)**

2K: ISP 空间是 **0x3400~0x3BFF(MPC89x51/52/53)**,**0xF400~0xFBFF(MPC89x54/58/515)**

1K: ISP 空间是 **0x3800~0x3BFF(MPC89x51/52/53)**,**0xF800~0xFBFF(MPC89x54/58/515)**

无: 无 ISP 空间 ，此时 IAP 空间也无效.

MOVCL: 用来决定 MOVC 指令是否有效

0: MOVC 有条件的无效

1: MOVC 总是有效

SB: 决定读出来的程序代码是否是乱码

0：是

1：否

LOCK: 决定是否对程序代码加锁

0: 是

1: 否

FZWDTCR: 用来锁定看门狗控制寄存器

0: WDTCR 寄存器仅在上电复位后被初始化成 0x00

1: WDTCR 寄存器在所有复位后都会初始化成 0x00(包括上电,RST 引脚,软件和看门狗)

OSCDN:

0: 如果频率小于 25MHz,该选项能用来减少内部增益来降低 EMI

1: 正常增益.

HWBS:

0: 上电后，系统从 ISP 空间启动

1: 上电后，系统从 AP 空间启动

EN6T:

0: MCU 运行在 6T 模式(每个机器周期为 6 clocks, 速度是标准 8051 的两倍)

1: MCU 运行在 12T 模式(每个机器周期为 12 clocks, 速度和标准 8051 的一样)

5 功能描述

5.1 定时/计数器

MPC89x5x 提供了 3 个 16 位定时/计数器 T0,T1,T2。每个都可以用来作为一般计数器。

当 **T0/T1/T2** 用作定时器时，用来触发定时器的时间单位是机器周期。一个机器周期等于 12 或 6 个振荡周期，这取决于用户配置这个芯片是 12T 模式，还是 6T 模式.

当 **T0/T1/T2** 用作计数器时，计数事件是 **T0/T1/T2** 对应引脚的“高到低的电平变化”。在这个模式，芯片每个机器周期对 **T0/T1/T2** 引脚进行采样。一旦结果是从 1 到 0,芯片就对计数器计 1。要注意的是,作为计数器操作时，T0/T1/T2 引脚的高脉冲或者低脉冲的宽度必须大于一个机器周期。

有两个特殊寄存器来配置 **T0** 和 **T1**.它们是 **TMOD,TCON**
有两个特殊寄存器来配置 **T2**.它们是 **T2MOD,T2CON**

■ **TMOD(0x89): TIMER 模式控制寄存器　　初始值　00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

用于 T1　　用于 T0

GATE:　0: 只要 TRx 置 1，Timer x 即使能
　　1: 必须 TRx 置 1，且/INTx 为高，Timer x 才使能
C/T:　0: 作为定时器
　　1: 作为计数器
M1,M0　模式选择
　　0,0:　作为 13 位定时/计数器
　　0,1:　作为 16 位定时/计数器
　　1,0:　作为 8 位自动重载定时/计数器，重载值存于 THx
　　1,1:　对于 T0, TL0 是一个 8 位定时/计数器, TH0 是一个 8 位定时器
　　T1 被停止

■ **TCON(0x88)　　初始值　00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1:　T1 溢出标志位
　　当 T1 溢出时，该位会自动置 1. 当执行 T1 溢出中断时，该位自动清零.
TR1:　0: 停止 T1
　　1: 开始 T1
TF0:　T0 溢出标志位
　　当 T0 溢出时，该位会自动置 1. 当执行 T0 溢出中断时，该位自动清零.
TR0:　0: 停止 T0
　　1: 开始 T0
IE1:　外部中断 1 标志
　　当外部中断 1 产生时，该位会自动置 1. 当执行外部中断 1 时，该位自动清零.

IT1: 0: 引脚 EX1 低电平，产生外部中断 0
1: 引脚 EX1 下降沿，产生外部中断 0
IE0: 外部中断 0 标志
当外部中断 0 产生时，该位会自动置 1. 当执行外部中断 0 时，该位自动清零.
IT0: 0: 引脚 EX0 低电平，产生外部中断 0
1: 引脚 EX0 下降沿，产生外部中断 0

■ **T2MOD(0xC9)** **初始值 XXXXXX00B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
-	-	-	-	-	-	T2OE	DCEN

T2OE: 定时器 2 输出使能位
0(默认): 禁止
1: T2 溢出输出到 P1.0(T2)
DCEN: 向下计数使能位
0(默认): 向上计数
1: 向下计数

■ **T2CON(0xC8)** **初始值 00000000B**

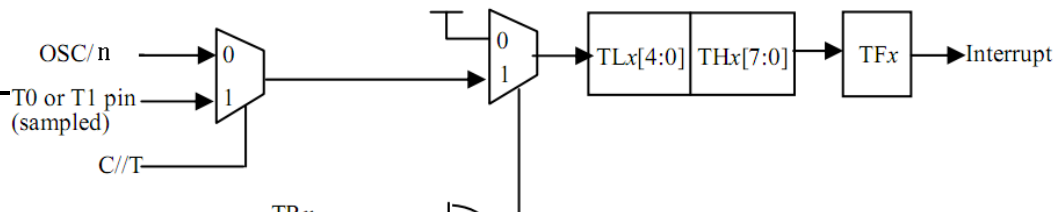
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

TF2 定时器 2 溢出标志。 定时器 2 溢出时置位.必须由软件清除 当 RCLK 或 TCLK=1 时 TF2 将不会置位。
EXF2 定时器 2 外部标志。当 EXEN2=1 且 T2EX 的负跳变产生捕获或重装时， EXF2 置位。定时器 2 中断使能时， EXF2 =1 将使 CPU 从中断向量处执行定时器 2 中断子程序。EXF2 位必须用软件清零 在递增/递减计数器模式(DCEN =1) 中， EXF2 不会引起中断。
RCLK 接收时钟标志。 RCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和模式 3 的接收时钟。 RCLK= 0 时，将定时器 1 的溢出脉冲作为接收时钟
TCLK 发送时钟标志。TCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和 模式 3 的发送时钟 TCLK= 0 时 将定时器 1 的溢出脉冲作为发送时钟
EXEN2 定时器 2 外部使能标志。当其置位且定时器 2 未作为串行口时钟时，允许 T2EX 的负跳变产生捕获或重装。 EXEN2= 0 时 T2EX 的跳变对定时器 2 无效。
TR2 定时器 2 启动/停止控制位。 置 1 时启动定时器。
C/T2 定时器/计数器选择。
0 内部定时器
1 外部事件计数器 下降沿触发
CP/RL2 捕获/重装标志。置位： EXEN2 =1 时 T2EX 的负跳变产生捕获。 清零：EXEN2= 1 时 定时器 2 溢出或 T2EX 的负跳变都可使定时器自动重装。 当 RCLK= 1 或 TCLK =1 时，该位无效且定时器强制为溢出时自动重装。

T0 和 T1 的四种模式

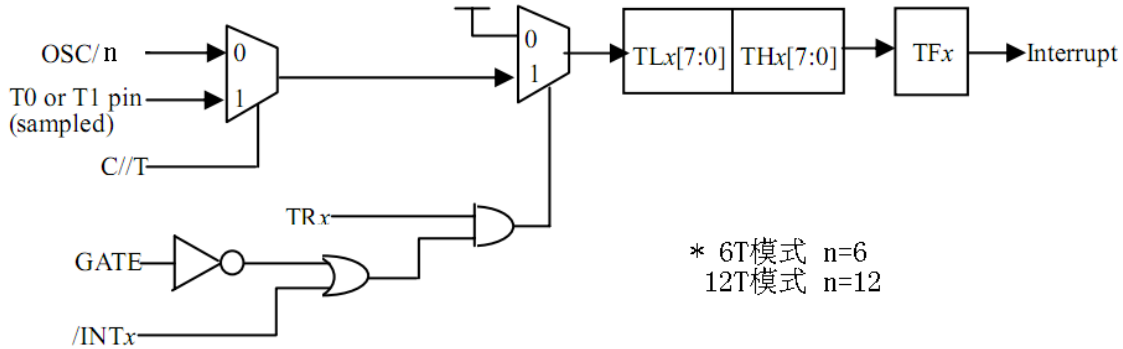
模式 0 M1,M0 = 0,0:

Timer 的寄存器被定义成 13 位寄存器。当寄存器由全 1 变成全 0 时，Timer 的中断标志位 **TFx** 将被置 1. 当 TRx=1 并且 GATE=0 或者 INTx=1 时，Timer 被使能。T0 和 T1 的模式 0 操作是一样的。



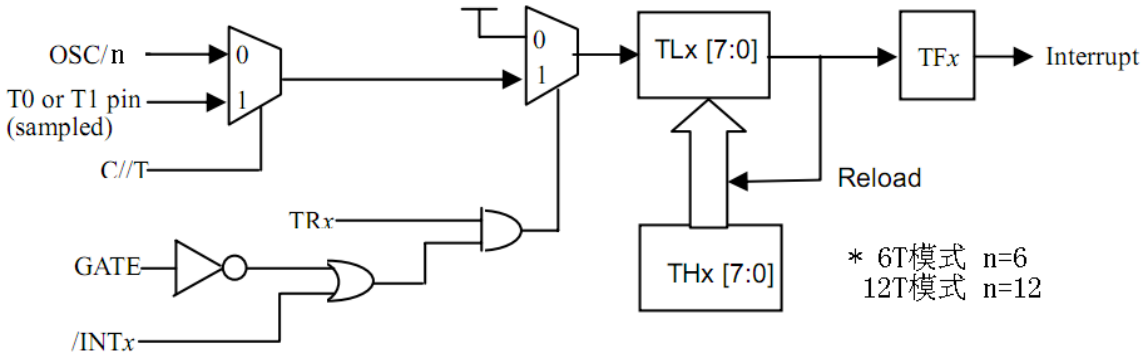
模式 1　M1,M0 = 0,1:

模式 1 除了是 16 位以外，其他的和模式 0 一样。



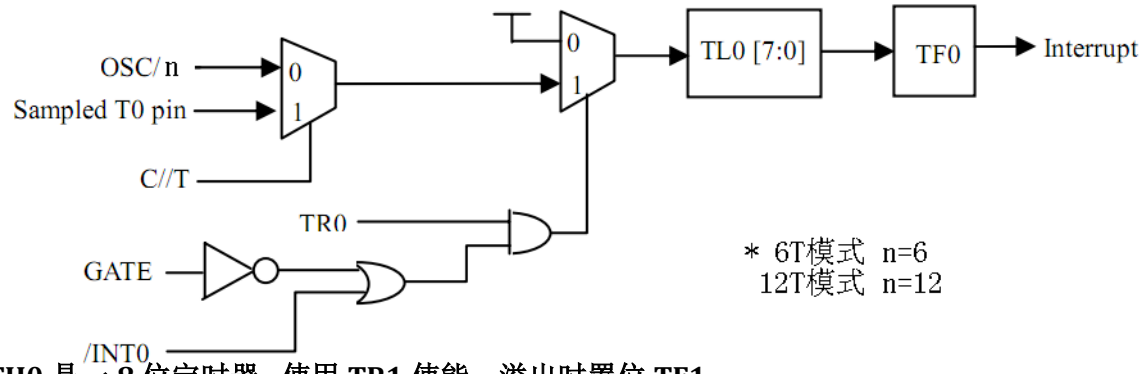
模式 2　M1,M0 = 1,0:

模式 2 配置成 8 位(TLx)自动重载计数器。TLx 溢出不仅置位 TFx,而且将 THx 的值加载到 TLx 中去，THx 的置由用户程序决定。重载后,THx 不会变。0 和 T1 的模式 0 操作是一样的。

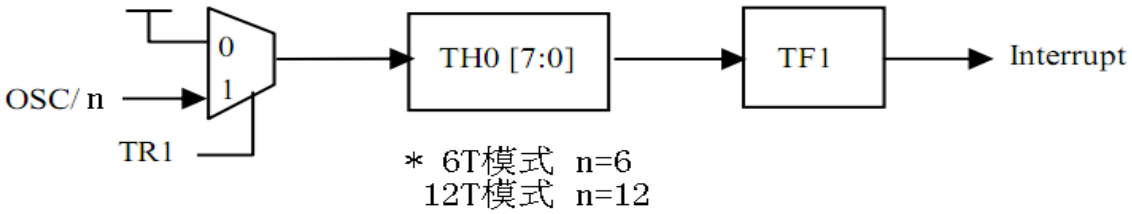


模式 3　M1,M0 = 1,1:

TL0 是一个 8 位定时/计数器



TH0 是一 8 位定时器，使用 TR1 使能，溢出时置位 TF1



T2 的四种模式

T2 是一个 16 位的定时器/计数器，可通过设置 **T2CON** 的 **C/T2** 位，让它工作在事件定时器或事件计数器。T2 有四种工作模式: 捕捉模式(CP),自动重载向下/向上模式(ARUD),自动重载向上模式(ARUO),波特率产生模式(BRG)

RCLK,TCLK	CP/RL2	TR2	DCEN	MODE
x	x	0	x	关闭
1	x	1	0	波特率产生模式(BRG)
0	1	1	0	捕捉模式(CP)
0	0	1	0	自动重载向上模式(ARUO)
0	0	1	1	自动重载向下/向上模式(ARUD)

Timer2 模式表

T2 也可设置成波形发生器。

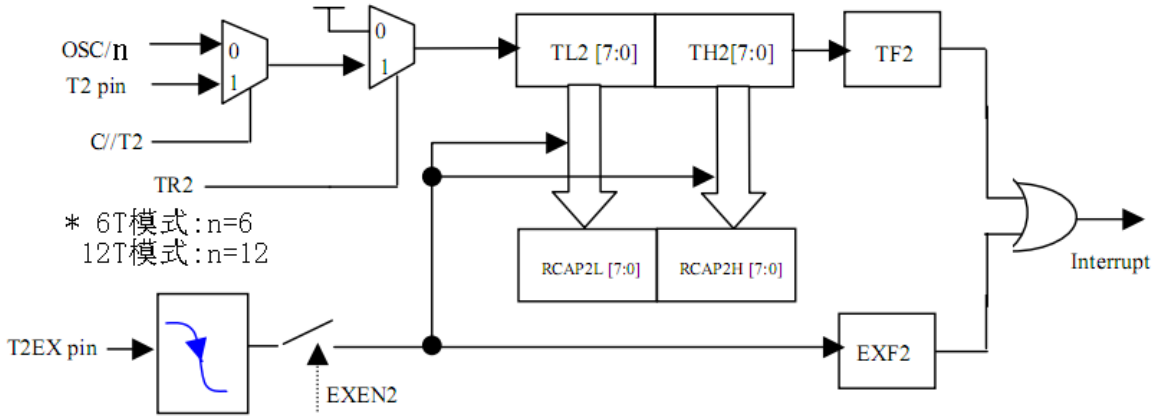
MPC89x5x 能够在 P1.0 口产生一个可编程时钟输出。当 T2OE 位被置 1,并且 C/T2 位清零，T2 溢出将在 P1.0 口输出一个占空比为 1:1 的脉冲波形。波形的频率由下列公式计算.

$$\frac{OSC}{nx(65536-RCAP2H,RCAP2L)}$$　(6T 模式: n=2;　12T 模式: n=4)

在时钟输出模式，Timer2 溢出将不会产生中断.

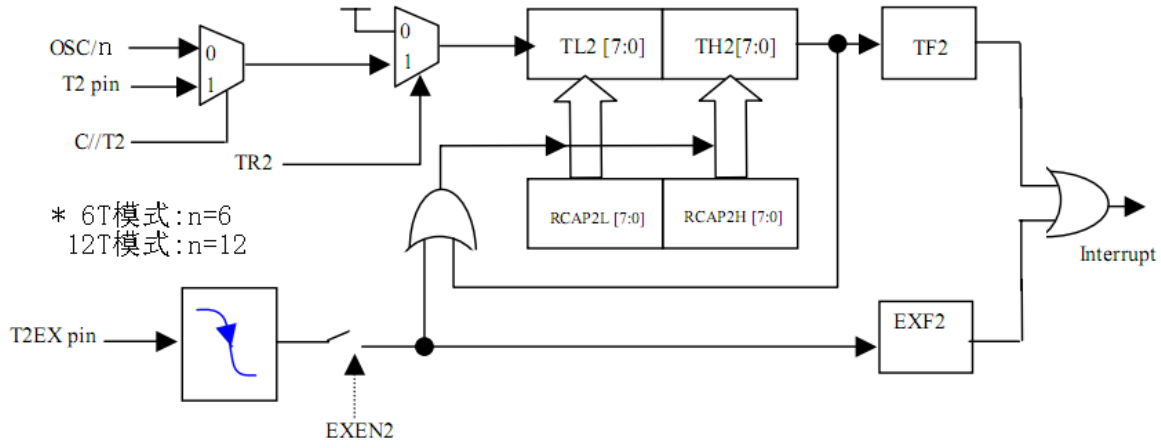
捕捉模式(CP)

作为捕捉模式，每 OSC/n(6T 模式:n=6,12T 模式:n=12)或外部引脚(T2)从高到低变化，T2 增加 1。**TR2** 使能 T2。 如果 **EXEN2=1**,当 **T2EX 引脚**产生**下降沿**，则会将 **T2** 的数值载入到 **RCAP2H:RCAP2L**。T2 溢出会置位 **TF2,** 如果 **EXEN2=1**, T2EX 引脚的下降沿会置位 **EXF2**。TF2 和 EXF2 都会产生 T2 中断。



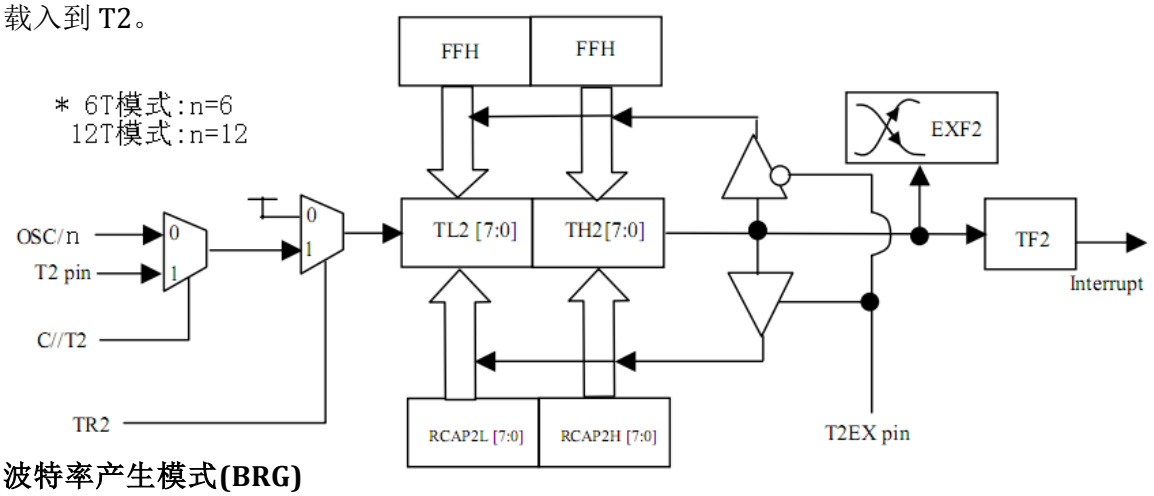
自动重载向上模式(ARUO)

作为 ARUO 模式，T2 被配置成向上增加，并可以由软件定义的值进行重载。复位后，DCEN=0 并且 CP/RL2=0,此时 T2 是 ARUO 模式。如果 **EXEN2=1**,当 **T2EX 引脚**产生**下降沿**，或者 T2 溢出，都会将 **RCAP2H:RCAP2L** 预设的数值载入到 **T2**。T2 溢出会置位 **TF2,** 如果 **EXEN2=1**, T2EX 引脚的下降沿会置位 **EXF2**。TF2 和 EXF2 都会产生 T2 中断。



自动重载向上/向下模式(ARUD)

作为 ARUD 模式，T2 被配置成可向上或向下。当 DCEN=1 并且 CP/RL2=0,此时 T2 是 ARUD 模式。**T2EX 引脚**的状态决定计数是向上还是向下。如果 T2EX=1,向上计数，否则向下计数。T2 溢出会置位 TF2 并且翻转 EXF2。在这个模式，EXF2 不能产生中断。如果计数方向是向下的，T2 溢出会将 0xFFFF 载入到 T2。如果计数方向是向上的，T2 溢出会将 RCAP2H:RCP2L 的数值载入到 T2。



波特率产生模式(BRG)

T2 可以设置成可变波特率发生器。T2CON 中的位 TCLK 和 RCLK 用来确定串口发送和接收的波特率的来源是 T1 还是 T2. 当 TCLK=0,T1 是串口发送的波特率来源。当 TCLK=1,T2 是串口发送的波特率来源。RCLK 有类似的功能。使用这两个位，串口发送和接收可以使用不相同的波特率——一个使用 T1，另一个使用 T2.

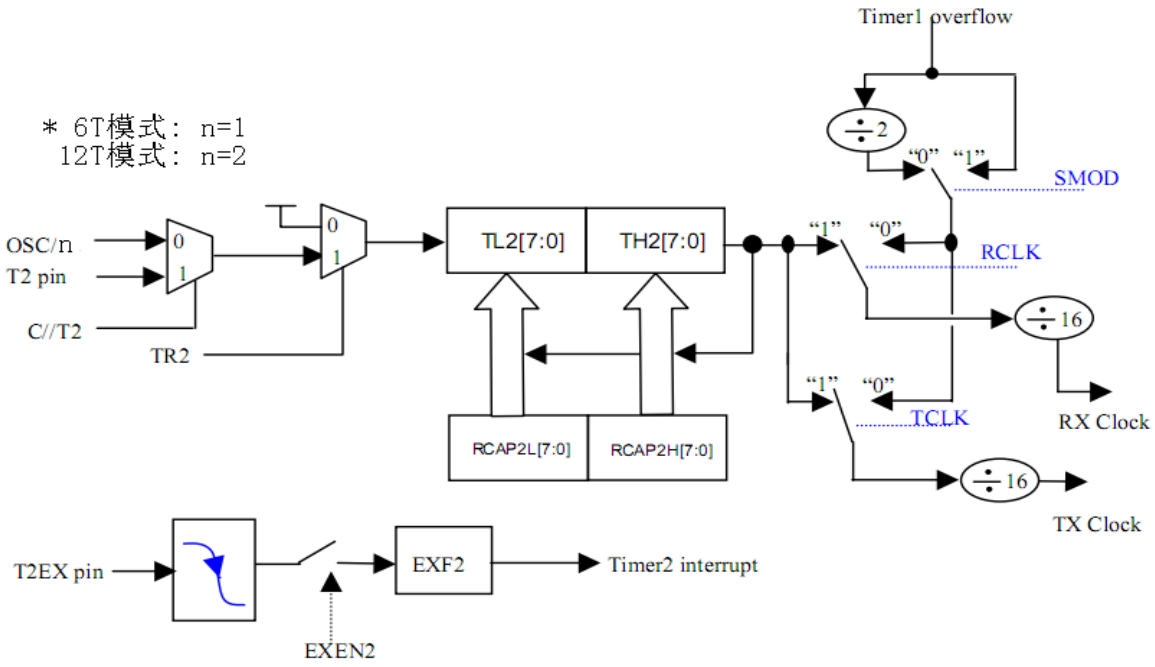
作为波特率发生器，T2 很像自动重载向上模式那样工作，除了 T2EX 引脚不能控制重载外。T2 溢出会将 RCAP2H:RCAP2L 的数值载入到 T2,但是 TF2 不会被置位产生中断。如果 EXEN2=1,T2EX 引脚下降沿会置位 EXF2 产生中断。

UART 模式 1 和模式 3 的波特率由 T2 溢出率决定，见下列公式:

波特率= T2 溢出率/16　　(计数 T2EX)

波特率= OSC/(16*n*(65536-RCAP2H:RCAP2L))　　(作为定时器)

*6T 模式: n = 1; 12T 模式: n=2;



5.1.1 定时器 0/1 示例

(1). 功能需求：定时器 T0 以 10KHz 的频率唤醒空闲模式，SYSCLK = 12MHz 晶振

汇编语言代码范例:

```
T0M0        EQU         01h
T0M1        EQU         02h
PT0     EQU         02h
PT0H        EQU         02h
IDL     EQU         01h

    ORG     0000h
    JMP     main

    ORG     0000Bh
time0_isr:
    to do…
    RETI

main:                                               ;
    MOV     TH0,#(256-100)              ; 设置定时器 0 溢出率为 = SYSCLK x 100
    MOV     TL0,#(256-100)              ;
    ANL     TMOD,#0F0h                  ; 设置定时器为模式 2
    ORL     TMOD,#T0M1                  ;
    CLR     TF0                             ; 清定时器 0 标志位

    ORL     IP,#PT0                         ; 选择定时器 0 中断优先级
    ORL     IPH,#PT0H                   ;

    SETB    ET0                             ; 使能定时器 0 中断
    SETB    EA                              ; 使能全局中断

    SETB    TR0                             ; 启动定时器 0 运行

    ORL     PCON,#IDL                   ; 设置 MCU 进入空闲模式
    JMP     $
```

C 语言代码范例:

```
#define T0M0            0x01
#define T0M1            0x02
#define PT0         0x02
#define PT0H            0x02
#define IDL         0x01

void time0_isr(void) interrupt 1
{
    To do…
```

```
}
void main(void)
{
    TH0 = TL0 = (256-100);                  //设置定时器 0 溢出率为 = SYSCLK x 100
    TMOD &= 0xF0;                           // S 设置定时器为模式 2
    TMOD |= T0M1;
    TF0 = 0;                                //清定时器 0 标志位

    IP |= PT0;                              //选择定时器 0 中断优先级
    IPH |= PT0H;
    ET0 = 1;                                //使能定时器 0 中断
    EA = 1;                                 //使能全局中断

    TR0 = 1;                                //启动定时器 0 运行
    PCON =IDL;                              //设置 MCU 进入空闲模式
    while(1);
}
```

(2). 功能需求：选择定时器 0 时钟源为 SYSCLK (使能 T0X12)

汇编语言代码范例:

```
T0M0        EQU         01h
T0M1        EQU         02h
PT0     EQU         02h
PT0H        EQU         02h
T0X12       EQU         80h

    ORG     0000h
    JMP     main

    ORG     0000Bh
time0_isr:
    to do…
    RETI

main:
    ORL     AUXR, #T0X12                ; 选择定时器 0 时钟源为 SYSCLK
    CLR     TF0                         ; 清定时器 0 标志位

    ORL     IP,#PT0                     ; 选择定时器 0 中断优先级
    ORL     IPH,#PT0H                   ;

    SETB    ET0                         ; 使能定时器 0 中断
    SETB    EA                          ; 使能全局中断

    MOV     TH0, #(256 - 240)           ;中断间隔 20us
    MOV     TL0, #(256 - 240)           ;

    ANL     TMOD,#0F0h                  ; 设置定时器 0 为模式 2
    ORL     TMOD,#T0M1                  ;


    SETB    TR0                         ; 启动定时器 0
    JMP     $
```

C 语言代码范例:

```
#define   T0M0          0x01
#define   T0M1          0x02
#define   PT0       0x02
#define   PT0H          0x02
#define   T0X12         0x80

    AUXR |= T0X12                       //选择定时器 0 时钟源为 SYSCLK
```

```
TF0 = 0;

IP |= PT0;                                  //选择定时器 0 中断优先级
IPH |= PT0H;

ET0 = 1;                                    //使能定时器 0 中断
EA = 1;                                     //使能全局中断

TH0 = TL0 = (256 - 240);

TMOD &= 0xF0;                               //设置定时器 0 为模式 2
TMOD |= T0M1;

TR0 = 1;                                    //启动定时器 0
```

5.2 中断

MPC89x5x 提供了 8 个中断源。每个中断源都可以通过特殊寄存器 **IE/XICON** 中的位来使能和禁止。该寄存器有一个 **EA** 位，清零它可以立刻禁止所有中断。

每个中断源都有两个对应的位来定义它的优先级。一个在 **IPH**，另一个在 **IP/XICON**。处理高优先级中断时，不会响应低优先级的中断请求。如果两个不同优先级的中断同时发出请求，高优先级的中断请求将会被响应。如果相同优先级的中断同时发出请求，则由内部优先级来决定哪个中断会被响应。下表说明了内部优先级和中断向量地址

中断源	中断向量地址	内部优先级
外部中断 0	03H	1(最高)
定时器 0	0BH	2
外部中断 1	13H	3
定时器 1	1BH	4
串口	23H	5
定时器 2	2BH	6
外部中断 2	33H	7
外部中断 3	3BH	8

外部中断**/INT0,/INT1,/INT2** 和**/INT3** 分别通过 **TCON** 的 **IT0**,**IT1,XICON** 的 **IT2, IT3** 可以设置成电平触发或边沿触发。实际产生的中断标志位是 **TCON** 的 **IE0**，**IE1**,**XICON** 的 **IE2** 和 **IE3**。产生外部中断时，如果是边沿触发，进入中断服务程序后由硬件清除中断标志位，如果中断是电平触发，由外部请求源而不是由片内硬件控制请求标志。

定时 0 和定时器 1 中断由 **TF0** 和 **TF1**(分别由各自的定时/计数寄存器控制，定时器 0 工作在模式 3 时除外)产生。当产生定时器中断时，进入中断服务程序后由片内硬件清除标志位。

串口中断由 **RI** 和 **TI** 的逻辑或产生。进入中断服务程序后，这些标志均不能被硬件清除。实际上，中断服务程序通常需要确定是由 RI 还是 TI 产生的中断，然后由软件清除中断标志。

定时器 2 中断由 **TF2** 和 **EXF**2 的逻辑或产生。进入中断服务程序后，这些标志均不能被硬件清除。实际上，中断服务程序通常需要确定是由 **TF2** 还是 **EXF**2 产生的中断，然后由软件清除中断标志。

所有这些产生中断的位都可通过软件置位或清零,与通过硬件置位或清零的效果相同。简而言之，中断可由软件产生，推迟或取消。

以下描述了中断的几个特殊寄存器:

■ **IE(0xA8)　　初始值: 00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

EA:　全局中断使能位
　　0:　禁止所有中断　　1:　使能中断

ET2:　定时器 2 中断使能位
　　0:　禁止　　1:　使能

ES:　串口中断使能位
　　0:　禁止　　1:　使能

ET1:　定时器 1 中断使能位
　　0:　禁止　　1:　使能

EX1:　外部中断使能位
　　0:　禁止　　1:　使能

ET0:　定时器 0 中断使能位
　　0:　禁止　　1:　使能

EX0: 外部中断 0 使能位

0: 禁止 1: 使能

■ **XICON(0xC0): 初始值: 00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2

PX3: 外部中断 3 优先级低位,与 IPH 中的 PX3H 一起使用

EX3: 外部中断 3 使能位

0: 禁止 1: 使能

IE3: 外部中断 3 中断标志位。外部中断 3 边沿触发时，有硬件置位，中断处理时由硬件清零

IT3: 外部中断 3 触发类型控制位

0: 边沿触发 1: 低电平触发

PX2: 外部中断 2 优先级低位,与 IPH 中的 PX2H 一起使用

EX2: 外部中断 2 使能位

0: 禁止 1: 使能

IE2: 外部中断 2 中断标志位。外部中断 3 边沿触发时，有硬件置位，中断处理时由硬件清零

IT2: 外部中断 2 触发类型控制位

0: 边沿触发 1: 低电平触发

■ **IP(0xB8) 中断优先级低位 初始值: 00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
---	---	PT2	PS	PT1	PX1	PT0	PX0

■ **IPH(0xB7) 中断优先级高位 初始值: 00000000B**

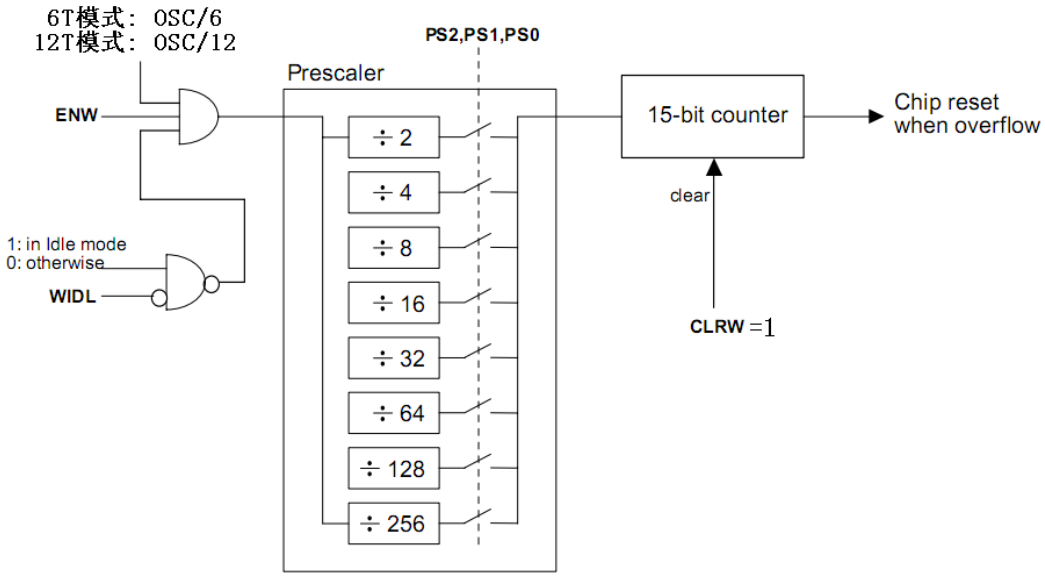
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
PX3H	PX2H	PT2	PSH	PT1H	PX1H	PT0H	PX0H

IP(或 XICON)和 IPH 一起决定了 4 级优先级，见下表

IPH.x,IP.x/XICON.x	优先级
1,1	1(最高)
1,0	2
0,1	3
0,0	4

5.3 看门狗

MPC82x52 提供了一个 15 位看门狗，8 位预分频。使能后，不能关闭。



■ **WDTCR(0xE1): 看门狗控制寄存器(只写)**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
WRF		ENW	CLRW	WIDL	PS2	PS1	PS0
上电复位		HWENW		HWIDL	HWPS2	HWPS1	HWPS0

WRF: 当 WDT 溢出时，该位被置 1。 由软件清零

ENW: 看门狗使能位。 上电后，该位由 **HWENW** 设定

1: 使能, (注意: 使能后软件不能关闭)

0: 禁止

CLRW: 对该位置 1， 将清零 WDT 计数器

WIDL: 上电后，该位由 **HWIDL** 设定

0: 在 IDLE 模式下停止 WDT 计数

1: 在 IDLE 模式下继续 WDT 计数

PS2,PS1,PS0 设置看门狗计数器预分频。上电后，该位由 **HWPS2:0** 设定

0,0,0	2
0,0,1	4
0,1,0	8
0,1,1	16
1,0,0	32
1,0,1	64
1,1,0	128
1,1,1	256

看门狗时间计算:

公式如下:

$$2^{15} \times (N \times \text{预分频} / Fosc)$$ **6T 模式：N=6; 12T 模式: N=12**

下表为 12Mhz 在不同预分频情况下的 WDT 溢出时间

PS2,PS1,PS0	预分频值	12T 模式	6T 模式
0,0,0	2	65.536ms	32.768ms
0,0,1	4	131.072ms	65.536ms
0,1,0	8	262.144ms	131.072ms
0,1,1	16	524.288ms	262.144ms
1,0,0	32	1.048s	524.288ms
1,0,1	64	2.097s	1.048s
1,1,0	128	4.194s	2.097s
1,1,1	256	8.389s	4.194s

5.3.1 定时器 0/1 示例

(1) *功能需求: 使能 WDT 并且选择 WDT 预分频为 1/32*

汇编语言代码范例:

```
PS0             EQU     01h
PS1             EQU     02h
PS2             EQU     04h
WIDL            EQU     08h
CLRW            EQU     10h
ENW             EQU     20h
WRF             EQU     80h

    ANL     WDTCR,#(0FFh - WRF)                 ; 清除 WRF 标志(写“0”)
    MOV     WDTCR,#(ENW + CLRW + PS2)   ; 使能 WDT 并且选择 WDT 预分频为 1/32
```

C 语言代码范例:

```
#define PS0     0x01
#define PS1     0x02
#define PS2     0x04
#define WIDL        0x08
#define CLRW            0x10
#define ENW     0x20
#define WRF     0x80

    WDTCR &= ~WRF;                              //清除 WRF 标志(写“0”)
    WDTCR = (ENW | CLRW | PS2);         //使能 WDT 并且选择 WDT 预分频为 1/32
                                        // PS[2:0]  | WDT 预分频器选项
                                        //  0   | 1/2
                                        //  1   | 1/4
                                        //  2   | 1/8
                                        //  3   | 1/16
                                        //  4   | 1/32
                                        //  5   | 1/64
                                        //  6   | 1/128
                                        //  7   | 1/256
```

5.4 串口(UART)

MPC89x5x 的串口是双工的，它可以同时进行收发数据。收发的数据共享同一个特殊寄存器 **SBUF**，实际在芯片内部是两个 **SBUF** 寄存器，一个用来发送，一个用来接收。该串口可工作在 4 个不同的模式。

模式 0

通常，这个模式纯粹是用来扩展芯片的 I/O 口。

在这个模式下，接收和发送数据都是通过 **RXD** 引脚，TXD 输出同步位移时钟。发送或接收的是 8 位数据，低位在先，其波特率固定为 MCU 时钟的 1/12。

模式 1

通过 TXD 引脚发送或者 RXD 引脚接收 10 位的数据帧。数据帧包括 1 个开始位(0),8 个数据位，1 个停止位(1)。接收完成后，停止位保存在 **SCON** 的 **RB8** 里。

波特率(模式 1) = 2^{SMOD} X (T1 溢出率) /32

或 = T2 溢出率 /16

模式 2

通过 TXD 引脚发送或者 RXD 引脚接收 11 位的数据帧。数据帧包括 1 个开始位(0),8 个数据位，1 个可编程第 9 位和 1 个停止位(1)。发送时，第 9 位来自 **SCON** 的 **TB8**。接收时，第 9 位数会存入到 **SCON** 的 **TB8**。波特率可选时钟频率的 1/32 或 1/64;

波特率(模式 2) = 2^{SMOD} X OSC/64

模式 3

模式 3 除了波特率与模式 2 不同外，其他都相同。

波特率(模式 3) = 2^{SMOD} X (T1 溢出率) /32

或 = T2 溢出率 /16

对所有模式，写 SBUF 都会启动发送动作。对于模式 0,如果 RI=0 并且 REN=1,则接收开始。对于其他模式，如果 REN=1 并且引入开始位，即有一下将沿，则接收开始。

以下描述了有关串口的几个特殊寄存器:

- **SBUF(0x99): 串口发送，接收数据寄存器 初始值：xxxxxxxxB**
- **SCON(0x98): 串口控制寄存器 初始值：00000000B**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

FE 帧错误位 当检测到一个无效停止位时 通过 UART 接收器设置该位 但它必须由软件清零 要使该位有效 PCON 寄存器中的 SMOD0 位必须置 1

SM0 和 SM1 定义串口操作模式 要使该位有效 PCON 寄存器中的 SMOD0 必须置 0

SM1 和 SM0 定义串行口操作模式 见下表

SM0	SM1	UART 模式	波特率
0	0	模式 0, 同步移位寄存器	osc/12
0	1	模式 1 ,8 位 UART	可变,取决于 Timer1 或 Timer2 溢出
1	0	模式 2, 9 位 UART	fosc /64 或 fosc /32
1	1	模式 3, 9 位 UART	可变,取决于 Timer1 或 Timer2 溢出

SM2 在模式 2 和 3 中多处理机通信使能位 在模式 2 或 3 中 若 SM2=1 且接收到的第 9 位数据 RB8 是 0 则 RI 接收中断标志 不会被激活 在模式 1 中 若 SM2=1 且没有接收到有效的停止位 则 RI 不会被激活 在模式 0 中 SM2 必须是 0

REN　允许接收位 由软件置位或清除 REN=1 时允许接收 REN=0 时 禁止接收

TB8　模式 2 和 3 中发送的第 9 位数据 可以按需要由软件置位或清除

RB8　模式 2 和 3 中已接收的第 9 位数据 在模式 1 中 或 sm2=0 RB8 是已接收的停止位 在模式 0 中 RB8 未用

TI　发送中断标志 模式 0 中 在发送完第 8 位数据时 由硬件置位 其它模式中 在发送停止位之初 由硬件置位 在任何模式中 都必须由软件来清除 TI

RI　接收中断标志 模式 0 中 接收第 8 位结束时由硬件置位 其它模式中在接收停止位的中间时刻 由硬件置位.在任何模式(SM2 所述情况除外)必须由软件清除

地址自动识别

有一增强功能可以轻松的做到一个主机，多个从机同步工作。它就是**地址自动识别。**

在芯片里有两个可读写的特殊寄存器 **SADDR** 和 **SADEN。**最终的，硬件由这两个寄存器“产生”一个“比较字”。公式如下:

Bit[i] of 比较字 = (SADEN[i]==1)?SADDR[i] : x

例如:

设 SADDR = 11000000B

SADEN = 11111101B

则：比较字 = 110000x0B　(x: 不考虑)

再如:

设 SADDR = 11100000B

SADEN = 11111010B

则：比较字 = 11100x0xB　(x: 不考虑)

产生比较字后，如果 MPC89x5x 收到一个 Byte，它会用“比较字”来比较这个 Byte 来确定是否要置位 **SCON** 的 **RI**。

一般的 UART 只要接收到一个 Byte 就会置位 **RI**。但是在 **MPC89x5x** 里，如果 **SCON** 的 **SM2**=1，它会按下列公式来设定 **RI**：

RI = (SM2==1)&&(SBUF==比较字)&&(RB8==1)

换句话说，不是所有的数据接收都会反应到 RI,除了一些特殊的数据外。

通过设定 **SADDR** 和 **SADEN**,用户可以筛掉那些他不想要的数据。此功能对于减小软件引导头有很大帮助。

以上功能适用于串口工作在模式 1，模式 2 和模式 3 下。

模式 0 下无效，用户可以忽略它。

帧错误检测

停止位丢失将会置位 **SCON** 的 **FE** 位。**SCON** 的 **FE** 位是与 **SM0** 位共享 **SCON.7** 的，实际上 **SCON.7** 的功能取决于 **PCON** 的 **SMOD0(PCON.6)**。如果 **SMOD**0=1,**SCON.7** 的功能是 **FE**,否则它的功能是 **SM0**。当作为 **FE** 位时，它只能由软件清零。

5.4.1 串行口示例

(1). 功能需求：串行口输入 RI 唤醒空闲模式

汇编语言代码范例：

```
PS              EQU             10h
PSH             EQU             10h



    ORG     00023h
uart_ri_idle_isr:
    JB      RI,RI_ISR                       ; 判断是否串行输入中断
    JB      TI,TI_ISR                       ; 判断是否串行发送中断
RETI                                        ; 中断返回

RI_ISR:
;   Process
    CLR     RI                              ; 清除 RI 标志
    RETI                                    ; 中断返回

TI_ISR:
;   Process
    CLR     TI                              ; 清除 TI 标志
    RETI                                    ; 中断返回

main:
    CLR     TI                              ; 清除 TI 标志
    CLR     RI                              ; 清除 RI 标志
    SETB    SM1                             ;
    SETB    REN                             ;8 位的模式 2，接收使能

    CALL    UART_Baud_Rate_Setting          ;参考 获得更多信息

    MOV     IP,#PSL                         ; 选择串行口中断优先级
    MOV     IPH,#PSH                        ;

    SETB    ES                              ; 使能串行口中断
    SETB    EA                              ; 使能全局中断

    ORL     PCON,#IDL;                      ; 设置 MCU 进入空闲模式
```

C 语言代码范例：

```
#define   PS              0x10
#define   PSH             0x10

void uart_ri_idle_isr(void) interrupt 4
{   if(RI)
```

```
    {
        RI=0;
        // to do ...
    }

    if(TI)
    {
        TI=0;
        // to do ...
    }
}

void main(void)
{
    TI = RI = 0;
    SM1 = REN = 1;                                  // 8 位的模式 2，接收使能

    UART_Baud_Rate_Setting()                    //参考 “ 获得更多信息

    IP = PSL;                                     //选择串行口中断优先级
    IPH = PSH;                                  //

    ES = 1;                                       // 使能串行口中断
    EA = 1;                                       //使能全局中断

    PCON |= IDL;                               //设置 MCU 进入空闲模式
}
```

5.5 复位

RESET 引脚用来复位芯片。它在芯片内部连接到一个施密特触发缓存，因此它能极好的去噪。在 **RESET 引脚**接**高电平**，必须大于**两个机器周期**，才能成功复位芯片。

5.6 省电模式和掉电模式

有两种节能模式，可以让 MPC89x5x 进入到节能模式。

IDLE 模式

用户置位 **PCON.0,**就可以让芯片进入到 **IDLE** 模式。

在 IDLE 模式，内部时钟与 CPU 断开，但是中断，定时器和串口还继续工作。

有两种方式可以结束 IDLE 模式。任何使能的中断的激活会导致 PCON.0 由硬件清零，而结束 IDLE 模式，中断被处理，紧接着是 RETI, 紧跟着设置进入 IDLE 模式指令的下一条指令将会被执行。另一种唤醒方法是，RESET 引脚接高，造成硬件复位。

掉电模式

用户置位 **PCON.1,**就可以让芯片进入到**掉电**模式。

在掉电模式下，芯片时钟停止。芯片内部 RAM 和 SFR 保持不变。

唤醒方法可以是**硬件复位**或**/INT0,/ITN1,/INT2 和/INT3 外部中断**。硬件复位时，注意至少要保持 RESET 引脚高电平超过 10ms 来稳定时钟, 程序从地址 0x0000 处开始执行。如果是外部中断唤醒，程序将会跳入中断向量，执行中断处理。为了使用外部中断唤醒，在进入掉电模式之前，必须正确设置外部中断。

注意在进入掉电模式指令的后面要加一条“NOP”指令

```
;*******************************************************************************************
;掉电模式唤醒示例(使用/INT0 来唤醒)
;*******************************************************************************************
INT0    BIT     0B2H        ;P3.2
EA      BIT     0AFH        ;IE.7
EX0     BIT     0A8H        ;IE.0

        CSEG    AT  0000h
        JMP     start
        CSEG    AT  0003h   ;外部中断 0 的中断向量
        JMP     IE0_isr
IE0_isr:
        CLR     EX0
        ;.....
        RETI;
Start:
        ;....
        SETB    INT0        ;拉高 P3.2
        CLR     IE0         ;清 INT0 中断标记
        SETB    IT0         ;选择下降沿触发
        SETB    EX0         ;使能外部中断 0
        SETB    EA          ;使能全局中断
        ORL     PCON,#02h   ;进入掉电模式
```

```
        ;...现在，MCU 进入到掉电模式，等待外部中断下降沿
        NOP                         ;注意，必须在这加一 NOP 指令
Wake_up:
        ;如果 INT0(P3.2)有一下降沿，则 MCU 会被唤醒，并进入”IE0_isr”,
        ;然后返回到这，继续运行.
```

IDLE 模式和掉电模式下的引脚状态

Mode	程序空间	ALE	PSEN	P0	P1	P2	P3
IDLE	内部	1	1	Data	Data	Data	Data
IDLE	外部	1	1	Float	Data	Address	Data
掉电模式	内部	0	0	Data	Data	Data	Data
掉电模式	外部	0	0	Float	Data	Data	Data

- **上电标志位(POF)**

  PCON.4 仅在上电的时候被置位，其它系统复位(看门狗，软件和 RESET 引脚)都不会置该位。它能由软件清零。

5.7 在系统编程(ISP)

为开发一个好 ISP 程序，用户应当了解内嵌 FLASH 的结构。

内嵌 FLASH 有 **30 页(MPC89x51/52/53),126 页(MPC89x54/58/515)**,每页有 **512 字节**。

处理 FLASH,用户在写数据之前必须先以页为单位进行擦除。擦除 FLASH 的意思是设定 FLASH 的数据为 FFh。在芯片里有两种擦除的方式，一种是整片擦除，另一种是页擦除。整片擦除有更好的效率，但它会擦除整片 FLASH.页擦除虽然没有那么高的效率，但是它更灵活。

不同于 RAM 的实时操作，擦除或写 FLASH 都有花比较长的时间。

此外，它有一个相当复杂的时序来处理擦除/编程 FLASH。幸运的是 MPC89x5x 提供便利的机制给用户读/写 FLASH 里的数据。仅仅添入数据和目标地址到几个特殊寄存器，然后触发内建的 ISP 自动操作，用户就可以轻松的擦除，读和写内嵌的 FLASH 和选项寄存器 **OR1**。

以下描述了有关 **ISP** 的几个特殊寄存器:

- **IFD(0xE2)**: ISP/IAP 操作的 数据
- **IFADRH(0xE3),IFDADRL(0xE4)**: ISP/IAP 操作的 地址
- **IFMT(0xE5)**: ISP/IAP 操作模式表
  =xxxxxx00 : 静态
  =xxxxxx01: 读数据
  =xxxxxx10: 写数据
  =xxxxxx11: 页面(512Bytes)擦除
- **SCMD(0xE6)**: ISP/IAP 操作时的命令触发寄存器，当顺序写入 0x46,0xB9 后，如果 ISP 使能(ISPCR.7 = 1),将会启动 ISP 操作.
- **ISPCR(0xE7):** ISP/IAP 控制命令寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ISPEN	SWBS	SWRST	CFAIL	---	WAIT2	WAIT1	WAIT0

ISPEN: 置 1 时，ISP 使能。

SWBS: 0: 芯片从 **AP** 空间启动;
1: 芯片从 **ISP** 空间启动;

SWRST: 置 1 时，芯片将会复位; 硬件自动清零

CFAIL: 上一次 ISP 操作结果标志
0: 成功； 1：失败

WAIT2,1,0: ISP 忙等待时间表

	CPU 等待时间(时钟周期)			
ISPCR.2:0	页擦除	写	读	对应系统时钟
000	43769	240	43	20~48M
001	21885	120	22	10~20M
010	10942	60	11	5~10M
011	5471	30	6	0~5M

ISP 基本操作(汇编)

```
;定义 IAP/ISP 命令即等待时间
ISP_READ      EQU     1           ;读
ISP_WRITE     EQU     2           ;写，字节为空(=0xFF)才能写进去
ISP_ERASE     EQU     3           ;页面(512Bytes)擦除，要某字节为空，只能擦除该字
                                  ;节所在的整个页面。
ISP_WAIT_TIME   EQU     1         ;设置等待时间。  具体数值参照“ISP 忙等待时间表”
                                  ;此处系统时钟为 12Mhz
```

字节读

```
MOV     IFADRH,     #BYTE_ADDR_H      ;送地址高字节
MOV     IFADRL,     #BYTE_ADDR_L      ;送地址低字节
CLR     EA                            ;关中断
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     ISPCR,      #ISP_WAIT_TIME    ;设置等待时间
ORL     ISPCR,      #10000000B        ;允许 ISP/IAP 操作
MOV     IFMT,       #ISP_READ         ;送读命令
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #46h              ;先送 46h
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #B9h              ;再送 B9h,  ISP/IAP 命令被触发启动
;CPU 等待 ISP/IAP 动作完成，此时程序被挂起。
NOP                                   ;
MOV     ISPCR,      #00000000B        ;清 ISP/IAP 特殊寄存器，防止误操作
SETB    EA          ;开中断
MOV     A,          IFD               ;将读出的数据送到 ACC
```

页面擦除

```
;擦除指定地址所在的页面。没有字节擦除，只能页面擦除，512 字节/页面，
;如果要对某个页面擦除，而其中有些字节又需要保留，则需在擦除前将其读到 RAM 中
;保存，然后再擦除页面，最后再将保存的数据写回该页面。
MOV     IFADRH,     #BYTE_ADDR_H      ;送地址高字节
MOV     IFADRL,     #BYTE_ADDR_L      ;送地址低字节
CLR     EA                            ;关中断
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     ISPCR,      #ISP_WAIT_TIME    ;设置等待时间
ORL     ISPCR,      #10000000B        ;允许 ISP/IAP 操作
MOV     IFMT,       #ISP_ERASE        ;送页面擦除命令
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #46h              ;先送 46h
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #B9h              ;再送 B9h,  ISP/IAP 命令被触发启动
;CPU 等待 ISP/IAP 动作完成，此时程序被挂起。
NOP                                   ;
MOV     ISPCR,      #00000000B        ;清 ISP/IAP 特殊寄存器，防止误操作
SETB    EA          ;开中断
```

写字节

```
;写字节到指定地址，该地址必须为空(0xFF),否则要先执行页面擦除
MOV     IFD,        #TEST_BYTE
MOV     IFADRH,     #BYTE_ADDR_H        ;送地址高字节
MOV     IFADRL,     #BYTE_ADDR_L        ;送地址低字节
CLR     EA                              ;关中断
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     ISPCR,      #ISP_WAIT_TIME      ;设置等待时间
ORL     ISPCR,      #10000000B          ;允许 ISP/IAP 操作
MOV     IFMT,       #ISP_WRITE          ;送页面擦除命令
;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #46h                ;先送 46h

;在这请加入软件陷阱判断，如出错，则不进行以下的操作了，可让芯片软复位
MOV     SCMD,       #B9h                ;再送 B9h,  ISP/IAP 命令被触发启动
;CPU 等待 ISP/IAP 动作完成，此时程序被挂起。
NOP                                     ;
MOV     ISPCR,      #00000000B          ;清 ISP/IAP 特殊寄存器，防止误操作
SETB    EA          ;开中断
```

程序启动入口

MPC89x5x 启动依照下列规则:

If (**HWBS==0**) && ({**ISPAS1,ISPAS0**}≠{**1,1**})

系统从 ISP 空间启动 ISP 程序

Else

系统从 AP 空间启动一般应用程序

以上规则仅在上电复位时有效，其它复位无效。

从 ISP 程序切换到 AP 应用程序

一旦 ISP 程序完成 FLASH 的数据更新，芯片就允许用户运行他的 AP 应用程序，只有在 ISP 程序的末尾加一条指令，如下:

ISPCR ← 001xxxxx

禁止写 FLASH,设置 SWBS=0，并且触发软件复位。然后系统就会复位(不是上电复位)，并且系统会检测 **SWBS=0**，从而在 AP 应用程序入口启动。对于上电过程，**HWBS** 将会决定程序的入口，但是软件复位的入口，是由 **SWBS** 决定的。

从 AP 应用程序切换到 ISP 程序

芯片允许用户应用程序切换到 ISP 程序，只有在应用程序中加一条指令，如下:

ISPCR ← x11xxxxx

设置 SWBS=1，从而选择软件复位的入口是 ISP 程序，并且触发软件复位。之后，系统就会复位(不是上电复位)，并且系统会检测 **SWBS=1**,从而在 ISP 程序入口启动。

5.8 在应用程序编程(IAP)

对于 MPC89x53/515，没有 IAP 功能。

IAP 是用来读/写非易失性 FLASH 数据的。有助于存储一些上电和掉电都需保存的参数。换句话说，用户能够存储数据到 FLASH 里去，然后掉电并且重启后，他能获取已经存储的原来的数据。

用户可以像 ISP 程序那样编程数据 FLASH，因此他要更深刻的了解特殊寄存器 **IFD**,**IFADRL**, **IFADRH**,**IFMT**,**SCMD** 和 **ISPCR**。

ISP 程序可以编程 AP 应用程序空间和数据 FLASH, 而 AP 应用程序可以编程数据 FLASH，但是不能编程 ISP 程序空间。如果 AP 应用程序要求改变 ISP 程序所在的特殊地址空间，硬件将会忽略这个请求。

6 额定极限(MPC89E51/52/53/54/58/515)

详细	范围
环境温度	-55~+125℃
存储温度	-65~150℃
任意 I/O 口或 RST 对地电压	-0.5~VCC+0.5V
VCC 对地电压	-0.5~+6.0V
VCC 到地最大电流	500mA
任意引脚最大灌电流	40mA

7 直流特性(MPC89E51/52/53/54/58/515)

不特别说明的情况下的条件是: VSS=0V, 环境温度=25℃ 12T 模式

标号	详细	测试环境	特性			单位
			最小	典型	最大	
V_{IL1}	输入低电平(P0,1,2,3,4)	Vcc=5.0V			0.8	V
V_{IL2}	输入低电平(RESET)	Vcc=5.0V			1.6	V
V_{IH1}	输入高电平(P0,1,2,3,4,EA)	Vcc=5.0V	2.0			V
V_{IH2}	输入高电平(RESET)	Vcc=5.0V	3.0			V
I_{OL1}	输出低时灌电流(P1,2,3,4)	Vcc=5.0V	4	6		mA
I_{OL2}	输出低时灌电流(P0,EA,PSEN)	Vcc=5.0V	8	12		mA
I_{OH1}	输出高时源出电流(P1,2,3,4)	Vcc=5.0V	150	220		uA
I_{OH2}	输出高时源电流(ALE,PSEN)	Vcc=5.0V	14	20		mA
I_{IL}	逻辑 0 输入电流(P1,2,3,4)	Vpin=0V		18	50	uA
I_{TL}	逻辑 1 到 0 转变电流(P1,2,3,4)	Vpin=2.0V		270	600	uA
I_{CC}	工作电流@20MHz	Vcc=5.0V			30	mA
I_{OH2}	IDLE 模式电流@20MHz	Vcc=5.0V			7	mA
I_{PD}	掉电模式电流	Vcc=5.0V			50	uA
Rrst	RESET 引脚内部下拉电阻		45		116	Kohm

8 额定极限(MPC89L51/52/53/54/58/515)

详细	范围
环境温度	-55~+125℃
存储温度	-65~150℃
任意 I/O 口或 RST 对地电压	-0.5~VCC+0.3V
VCC 对地电压	-0.3~+4.20V
VCC 到地最大电流	500mA
任意引脚最大灌电流	40mA

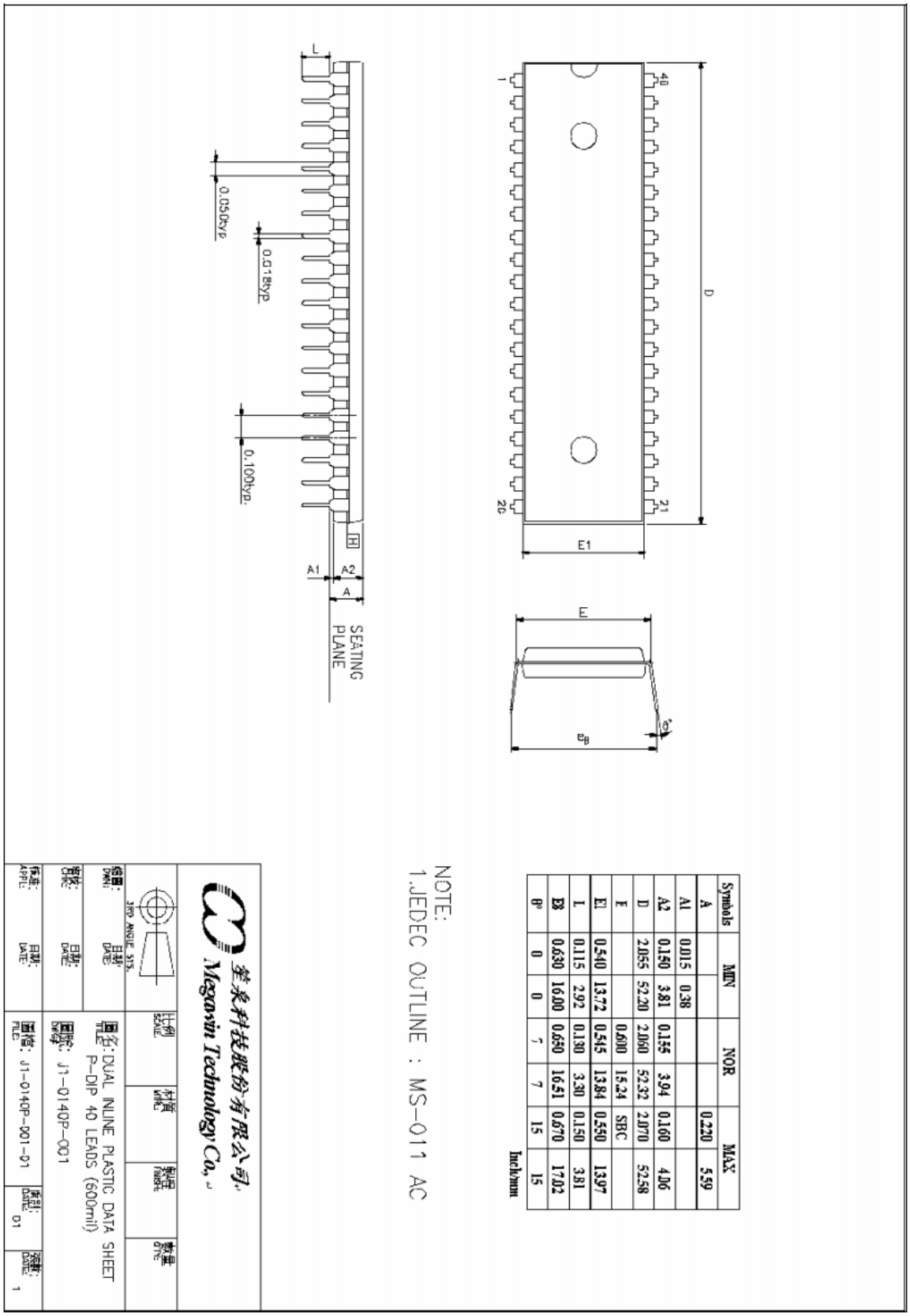
9 直流特性(MPC89L51/52/53/54/58/515)

不特别说明的情况下的条件是: VSS=0V, 环境温度=25℃ 12T 模式

标号	详细	测试环境	特性			单位
			最小	典型	最大	
V_{IL1}	输入低电平(P0,1,2,3,4)	Vcc=3.3V			0.8	V
V_{IL2}	输入低电平(RESET)	Vcc=3.3V			1.5	V
V_{IH1}	输入高电平(P0,1,2,3,4,EA)	Vcc=3.3V	2.0			V
V_{IH2}	输入高电平(RESET)	Vcc=3.3V	3.0			V
I_{OL1}	输出低时灌电流(P1,2,3,4)	Vcc=3.3V	2.5	4		mA
I_{OL2}	输出低时灌电流(P0,EA,PSEN)	Vcc=3.3V	5	8		mA
I_{OH1}	输出高时源出电流(P1,2,3,4)	Vcc=3.3V	40	70		uA
I_{OH2}	输出高时源电流(ALE,PSEN)	Vcc=3.3V	8	13		mA
I_{IL}	逻辑 0 输入电流(P1,2,3,4)	Vpin=0V		8	50	uA
I_{TL}	逻辑 1 到 0 转变电流(P1,2,3,4)	Vpin=2.0V		110	600	uA
I_{CC}	工作电流@20MHz	Vcc=3.3V			30	mA
I_{OH2}	IDLE 模式电流@20MHz	Vcc=3.3V			6	mA
I_{PD}	掉电模式电流	Vcc=3.3V			50	uA
Rrst	RESET 引脚内部下拉电阻		45		116	Kohm

10 封装尺寸

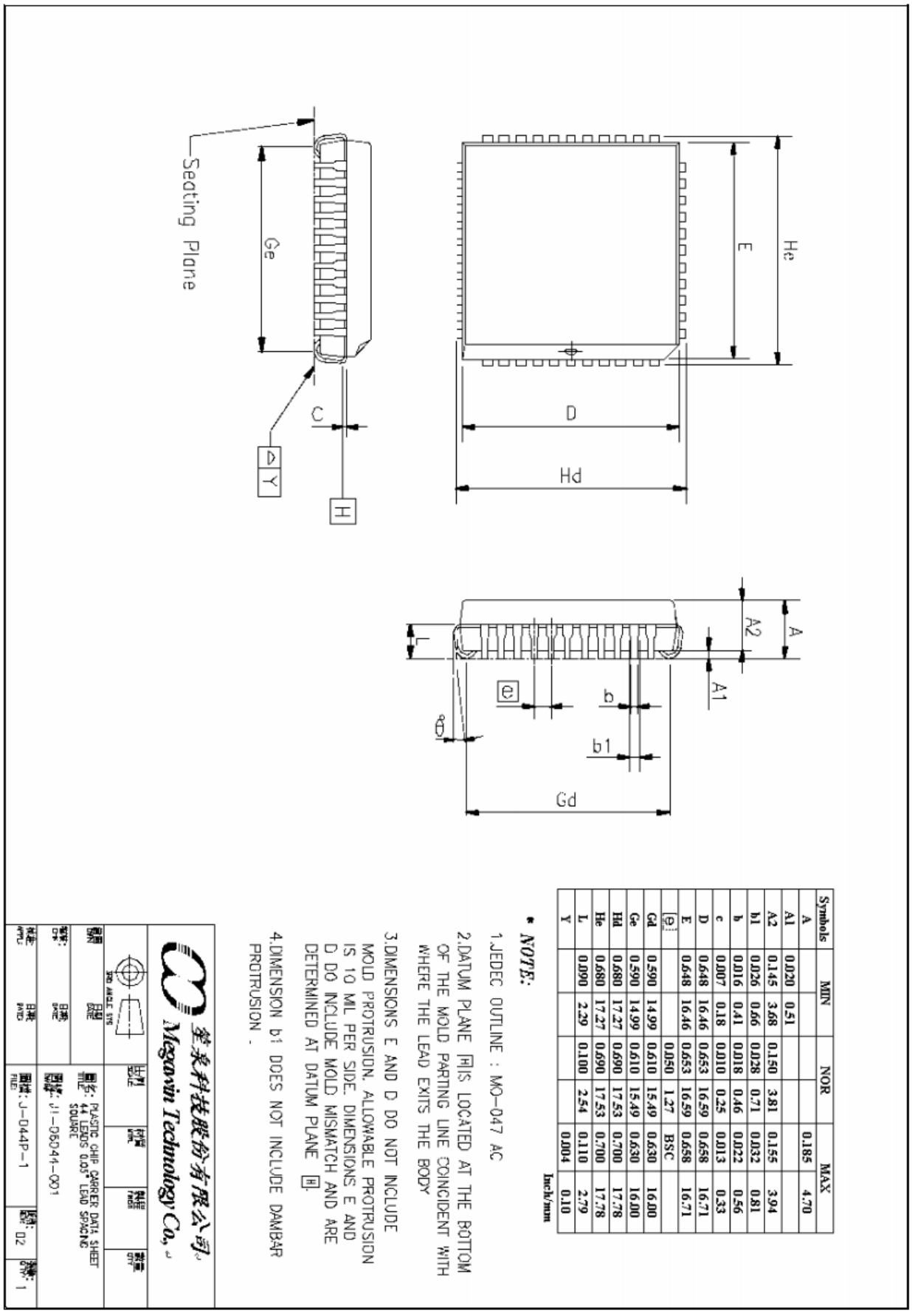
40PIN PDIP



Symbols	MIN	NOR	MAX
A			0.220
A1	0.015	0.38	
A2	0.150	3.81	0.155	3.94	0.160	4.06
D	2.055	52.20	2.060	52.32	2.070	52.58
F		0.600	15.24	SBC	
E1	0.540	13.72	0.545	13.84	0.550	13.97
L	0.115	2.92	0.130	3.30	0.150	3.81
eB	0.630	16.00	0.650	16.51	0.670	17.02
θ°	0	0	7	7	15	15

Inch/mm

44PIN PLCC



Symbols	MIN	NOR	MAX
A			4.70
A1	0.020	0.51	
A2	0.145	3.68	0.150	3.81	0.155	3.94
b1	0.026	0.66	0.028	0.71	0.032	0.81
b	0.016	0.41	0.018	0.46	0.022	0.56
c	0.007	0.18	0.010	0.25	0.013	0.33
D	0.648	16.46	0.653	16.59	0.658	16.71
E	0.648	16.46	0.653	16.59	0.658	16.71
e			0.050	1.27	BSC	
Gd	0.590	14.99	0.610	15.49	0.630	16.00
Ge	0.590	14.99	0.610	15.49	0.630	16.00
Hd	0.680	17.27	0.690	17.53	0.700	17.78
He	0.680	17.27	0.690	17.53	0.700	17.78
L	0.090	2.29	0.100	2.54	0.110	2.79
Y					0.004	0.10

Inch/mm

*** *NOTE:***

1.JEDEC OUTLINE : MO-047 AC

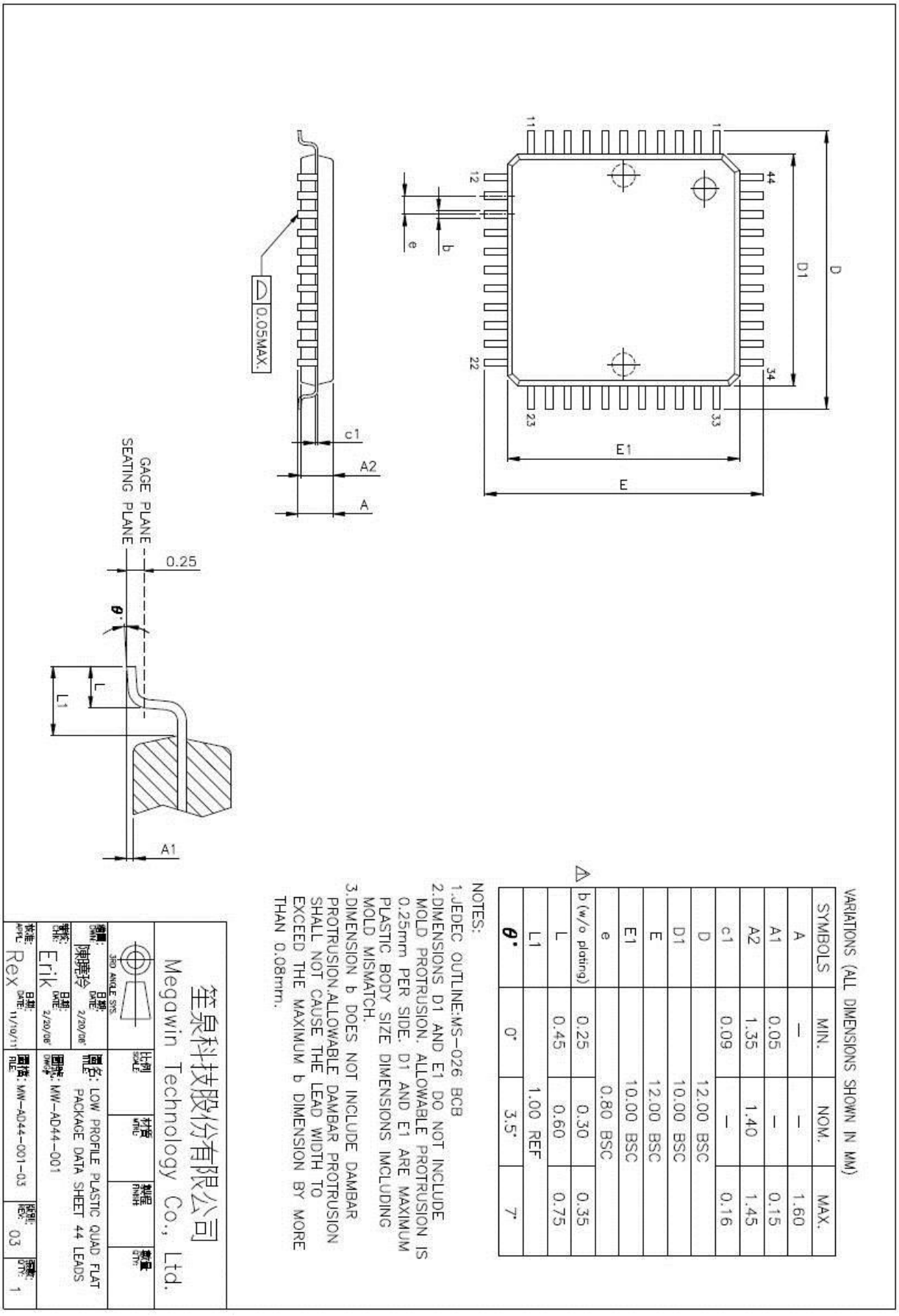
2.DATUM PLANE [H]IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY

3.DIMENSIONS E AND D DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 10 MIL PER SIDE. DIMENSIONS E AND D DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE [H].

4.DIMENSION b1 DOES NOT INCLUDE DAMBAR PROTRUSION .

麦来科技股份有限公司 Megawin Technology Co.,			
3RD ANGLE VIEW	比例 SCALE	材料 MATL	数量 QTY
品名: PLASTIC CHIP CARRIER DATA SHEET 44 LEADS 0.05" LEAD SPACING SQUARE			
制图: DRWG	日期: DATE	图号: J1-D5044-001	
审核: APPD	日期: DATE	图档: J-D44P-1	版本: D2
		FILE	REV

44PIN LQFP



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
c1	0.09	—	0.16
D		12.00 BSC	
D1		10.00 BSC	
E		12.00 BSC	
E1		10.00 BSC	
e		0.80 BSC	
b (w/o plating)	0.25	0.30	0.35
L	0.45	0.60	0.75
L1		1.00 REF	
θ°	0°	3.5°	7°

NOTES:
1.JEDEC OUTLINE:MS-026 BCB
2.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS IMCLUDING MOLD MISMATCH.
3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.

笙泉科技股份有限公司
Megawin Technology Co., Ltd.

3RD ANGLE SYS	比例 SCALE:	材質 MTRL:	數量 QTY:
繪圖 DWN: 陳勝玲 DATE: 2/20/08	品名: LOW PROFILE PLASTIC QUAD FLAT PACKAGE DATA SHEET 44 LEADS		
設計 CHK: Erik DATE: 2/20/08	圖號: MW-AD44-001		
核准 APPL: Rex DATE: 11/10/11	檔名: MW-AD44-001-03	版號 REV: 03	張次 QTY: 1

11 修订历史

版本	日期	描述
V1.00	2009.02.25	初次版本
V1.01	2015.01.06	新增 LQFP-44 封装，移除 PQFP-44 封装
V1.02	2015.09.25	修正章节” **5.7 在系统编程(ISP)** ”部份内容