

FOCJ1寻址.

Rn 8位通用寄存器, n=0~7

Ri 可用中间寄存器的寄存器, i=0,1

#data #13立即数

#data16 #16立即数.

:

寻址方式及空间.

1. 立即寻址.

MOV A, #3AH //立即数前加#.

2. 直接寻址.

MOV A, 3AH //将3AH地址内的数据放入A

3. 寄存器寻址.

INC R0 //R0的值+1再赋于A.

4. 寄存器间接寻址.(只能用R0,R1)

MOV A, @Ri (i=0,1) //将Ri内容作为地址,取该地址的数据.

5. 相对寻址.

6. 变址寻址

(DPTR用于访问片外数据存储器, 16位).

MOV C A, @A+DPTR

MOV C A, @A+PC

JMP @A+DPTR.

7. 16位寻址.

片内RAM中20H~2FH中16个单元1813, 编码为00H~7FH,

例 MOV C, 2BH.

MOV <目标字节>, <源字节>.

进位和 PC1 算术.

4T 的空间. { 内部数据存储器.

| 内部数据存储器.

3T 的空间. { 64K 程序存储器. 0000 ~ FFFFH

| 64K 外部数据存储器 0000 ~ FFFFH.

| 256B 内部数据存储器. 00 ~ FFH

1) 程序存储器.

外部 $\begin{array}{c} \text{FFFFH} \\ | \\ 100011 \end{array}$

$\begin{array}{c} 0FFFH \\ | \\ 000011 \end{array}$ 内部 $\begin{array}{c} =EA= \\ | \rightarrow | \\ 61.61S \\ | \\ 0FFFH \\ | \\ 000011 \end{array}$

保留单元: 000 ~ 00024H, 程序起始, 一般设置为可跳转地址.

内部 4KB 存储

空间即 0FFFH

eg: LJMP 0050H

0003 ~ 002AH. 5 个中断.

0003H: 外部中断 0

000BH: 定时器 0 溢出.

0013H: 外部中断 1

001BH: 定时器 1 溢出

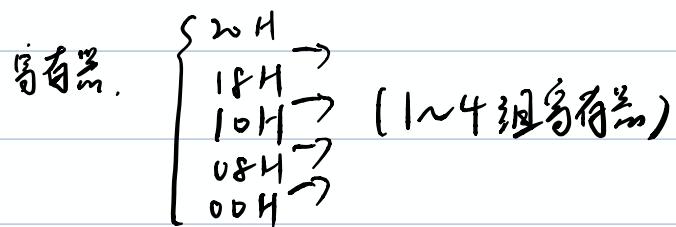
0023H: 单片机中断.

内部数据存储器.

256字节: 00H~FFH |
 | 128字节 00H~7FH: 内部数据 RAM
 | 128 80H~FFH: 特殊功能寄存器.

80C52 中 256B RAM, 80-FFH SFR 与 RAM 重合.

RAM 分配.



7 组地址 20H → 2FH → 16 个存储空间 表存区 ← 128 位 00H ~ 7FH.
字节寻址: 30H → 7FH.

4 组 8 位寄存器, 由 PSW 中 R0, R1 选择.

寻址 (73) RAM 区

MOV A, 40H // 将 40H 的 8 位数据 → A

MOV C, 40H // 将 16 地址为 40H (寻址地址为 28H 的 073) 送入 C

3. 特殊寄存器 (SFR) 80H~FFH

21 个特殊功能寄存器.

80C52 中 内部 RAM 与 SFR 重合, 通过不同寻址方式 区别访问.

{ RAM, 间接寻址.
SFR, 直接寻址

1) 程序寄存器 PC, 用于存放下一寻址地址.

2) 寄存器 A (R0H)

3) 寄存器 B (R0H)

4) 程序状态字寄存器 PSW (R0H), 包含寻址.

5) I/O 寄存器

P0 80H P1 90H

P2 A0H P3 B0H

均同理.

6) 堆栈指针 SP (R1H)

7) 数据指针 DPTR { DPH f3H
| bPL f2H

DPTR 可以由 16 位也可由 8 位组成.

4. 外部数据存储器. 最大容量: 64K. 0000H ~ FFFFH.

16 位数据指针

DPTR 或 R0, R1 组成.

传送指令.

MOV A, #data (A) ← #data

MOV Rn, #data (Rn) ← #data

MOV @Ri #data ((Ri)) ← #data (i=0,1)

分支 LJMP, AJMP, SJMP, JMP ← 跳转语句.

↑ ↑ ↓

长转移(64K) 短转移(2KB) 循环转移(+16+19)

绪论部分

什么是嵌入式：以应用为中心，以计算机技术为基础的计算机系统
针对具体应用的专用计算机系统。

硬件单元：CPU、程序存储器、数据存储器、SFR、4个并行I/O口、2个定时器
↑4KB ↑128B +5个中断源。

CPU时序：1. 存储周期

2. 机器周期（2个存储周期）。

3. 指令周期（1~2个机器周期）

若晶振为12MHz，则机器周期为1μs，指令为1μs、2μs、4μs。

数据存储：哈佛结构，程序与数据分开编址。

- 4个物理空间（内、外X（数据、程序））
- 3个逻辑空间

程序存储器：0000H ~ FFFFH ($2^{16} \rightarrow 64\text{ KB}$)

↑ 0000H ~ 1000H 由 RAM 采用内外存储器。

B寄存器区（中断地址等）

数据存储器：00 ~ 7FH → RAM → {00 ~ 20H 4组寄存器，每组8个，每个8位。
20 ~ 2FH 可直接寻址。

↑ MOV A <AH <字节

Mov C, xxH <寻址。

7/012 → \$0H 90H A0H B0H, 问句待回答.

P1 P2 P3 P4. ↑

堆栈指针: 81H

P3 功能复用

3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7

基本寻址方式: RxD TXD \overline{RD} \overline{WD} T₀ T₁ \overline{WR} \overline{RD}

1. 寄存器寻址 { R₀ ~ R₇

ACC

B, DPTR, 位寄存器, Cy.

2. 直接寻址 { 00~7FH

SFR

位地址空间

3. 寄存器间接寻址: (可用 R₀, R₁, DPTR, SP) 备号用 @.

4. 立即寻址: 需要在立即数前面十进制.

5. 相对寻址:

6. 变址间接寻址: { 基址: PL, DPTR, MOV C A, @A+DPTR.

变址: A

7. 位寻址.

传送指令: 1. 立即数. MOV A, #Data

MOV Rh, #Data $\leftarrow (Rh) \leftarrow Data$

MOV @Ri, #Data $\leftarrow ((Ri)) \leftarrow Data$

放入寄存器所指向的空间.

Mov. direct #Data. { direct: 00~FFH.

↑由 00~7FH 及 SFR.

SFR 不可使用间接寻址方式.

外部数据：只能间接寻址，且不能与累加器A进行数据传送。

指令：MOVX

$\left\{ \begin{array}{l} \text{MOV DPTR, \#300H} \\ \text{MOVX A, @DPTR.} \end{array} \right.$



$\left\{ \begin{array}{l} \text{MOV P2, \#30H} \\ \text{MOV R0, \#07H} \\ \text{MOV X A, @R0} \end{array} \right.$

P2为向量地址，R0为存放地址。
进行寻址。

MOV C A, @A+PC/DPTR.

XCH：字节交换指令。

XCHD：双字节交换指令。

SWAP：累加器A向低四位交换位。

算术类：ADD, ADDC, SUBB. 计算时先 + / - C P P0.
将会影响标志位 CPSW 中的 OV, CY, AZ, P.

ADD : $A \leftarrow B + A$

$\left\{ \begin{array}{l} \text{有进位} CY = 1 \\ \text{3至4有进位} AC = 1 \xrightarrow{\text{变化相同.}} \\ 6.7位无进位} OV = 0, \text{否则} OV = 1 \\ A \text{中有借位} Z = 1, P = 0, \text{否则} P = 1 \end{array} \right.$

0C3H + 084H

$$\begin{array}{r}
 1100\ 0011 \\
 1010\ 1010 \\
 \hline
 0110\ 1101
 \end{array}
 \quad
 \begin{array}{l}
 Cy=1 \\
 Ac=0 \\
 N=1 \\
 P=1
 \end{array}$$

SUBB: $A \leftarrow A - B$.

最高位借位: $Cy=1$

最低位借位: $Ac=1$

最高两位相同: $NV=1$ [否则] = 0

$(A) = 0C1H$, $(B) = 54H$, $C=1$

SUBB A, B

1100 1001

00100

乘法指令: MUL. AB

若积大于255, NV=1, C总被清零

$(A) = 80H (128) \times B = 32H \text{ (NO)}$

MUL AB \rightarrow (A) 00H (B) = 32H

$NV=1$

$Cy=1$

$P=0$

除法指令: DIV AB; $\begin{cases} (A) \leftarrow 商 \\ (B) \leftarrow 余数 \end{cases} \rightarrow (A) / B$

CLR : 清零.

CPL : 取反

RL : 左移左移.

RRC : 右进位右移右移

ANL : 逻辑与. 放入A. ← 常用值0、与0相与.

ORL : 逻辑或 ← 常用值111、与1相或

XRL : 逻辑异或 ← 常用于取反、与1或取反

转移类指令

1. 绝对无条件转移：AJMP，范围有PC.

2. 相对转移指令：LJMP addr16

3. 相对转移指令：SJMP rel

4. 间接转移：JMP @A+DPTR.

条件转移：

1. JZ, JNZ 没零 (判断A是否为0)

循环转移：

1. CJNE A, #Data, L0

↑ 不等于、相等不转移.

循环转移：

1. DJNZ Rn, rel

↑ 减1并循环，直至Rn=0.

子程序调用：LCALL addr16

绝对调用：ACALL addr16

中断返回：RETI [A同RETI]. B. 恢复中断连接，使CPU可再接收中断

位传送指令 (用C)

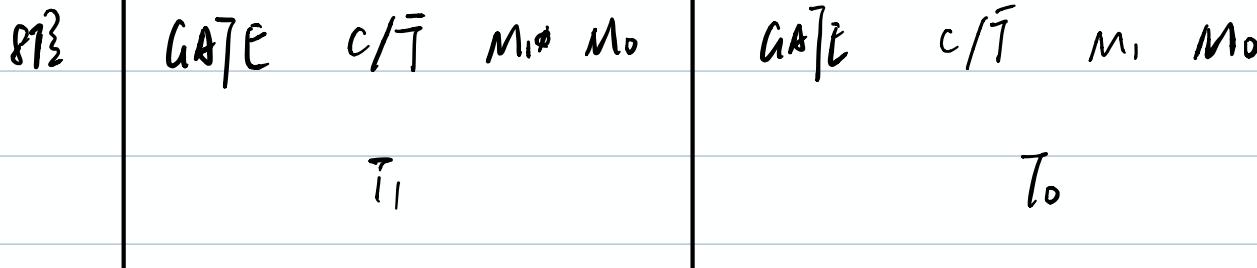
MOV C, 30H

MOV 20H, C (30H) → (20H) 1字节/2.

CLR, CPL, SETB
 ↑清零 ↑取反 ↑置位

延时器: SFR

TMOD: 89H



00: 13H 01: 16H 10: 自动重装89H. 11: 分为两个89H.

GATE: 选通通道. 除TRX=1外, TR1和TR0为高才计数. 用于测脉宽用.

TCON: 88H

TF, TR, TF₀, TR₀, TI, IE1, IE0, IT0

TF: 溢出中断请求.

TI: 0为边沿触发, 内电平触发.

IE: 外部中断D

初始化顺序:

TMOD → TH, TL → ET, EA → TRX

计算延时时间:

↓机器周期, TP = 12/fosc.

$$TC = X \cdot TP$$

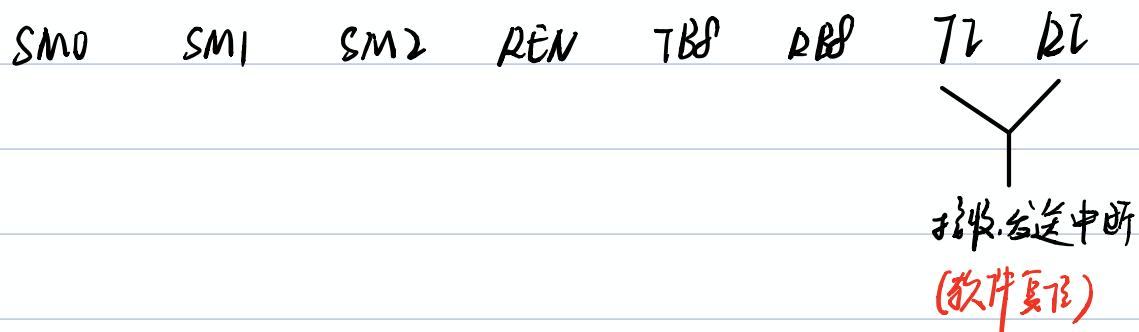
↑高电平时, ↑计数个数

$$X = TC / TP$$

计数初值: 2ⁿ - X. (看工作方式决定n).

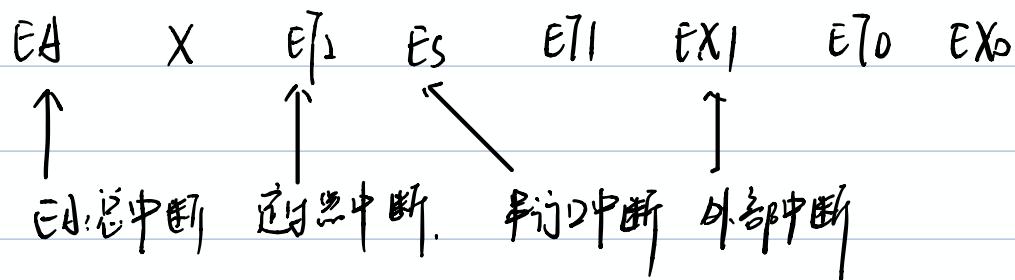
串行控制寄存器.

SCON : 98H.



中断允许控制位:

IE : 48H



中断源矢量地址

INT0 0003H

T0 0013H

INT1 0013H

T1 001BH

串口1 0023H

T2 002BH.

<产生复位时 P₀, P₁, P₂, P₃初值为1, SP复位为07H>.

串行通讯

收发器共用一个地址码 → 99H

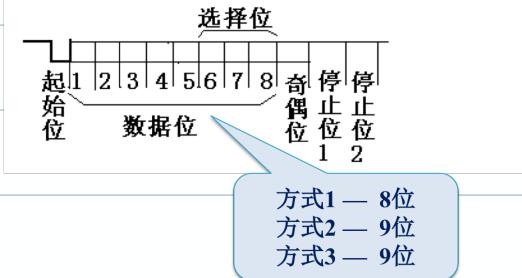
SCON

SM0 SM1 SM2 REN TBS DB8 TI DL

PCON : PCON.7 \Rightarrow SMOD. 声明波特率加倍.

故对注意REN=1

串行工作方式



波特率设置

式1: $f_{osc}/12$.

式2: $2^{SMOD} \times f_{osc}/64$

式1.3: $2^{SMOD} \times (T1\text{溢出})/32$

$\uparrow = \text{定时器溢出次数/秒}$

$= 1/\text{溢出周期}$.

一般采用T1的式2

$$T1\text{溢出} = \frac{f_{osc}}{12} \times \frac{1}{2^{8-(7H1)}}$$

$$\text{波特率} = 2^{SMOD} \times (T1\text{溢出})/32$$

注意禁用T1中断

$$N = 256 - \frac{2^{SMOD} \times f_{osc}}{12 \times 32 \times \text{波特率}}$$

$f_{osc} = 6MHz$, $SMOD = 1$, 波特率 = 240bps

$$N = 256 - \frac{2 \times 6 \times 10^6}{12 \times 32 \times 240} \approx 243 = 134$$

老式 |

◆ 常用波特率与定时/计数器1各参数关系 (串行方式1和3)

波特率	振荡频率	SMOD	T1方式	重新装入值
62.5k	12MHZ	1	2	FFH
19.2k	11.059MHZ	1	2	FDH
9.6k	11.059MHZ	0	2	FDH
4.8k	11.059MHZ	0	2	FAH
2.4k	11.059MHZ	0	2	F4H
1.2k	11.059MHZ	0	2	E8H
137.5	11.059MHZ	0	2	1DH
110	6MHZ	0	2	72H

· 可扩展系统。

80C51可扩展为提高存储器。

控制总线：

ALE：访问外部地址时，ALE负跳变将P0口子线送入地址锁存器。

PSEN：

WR、RD：读写选通

对于732，P₀提供低8位地址，P₁提供高四位地址。

ADC 0f04 读写时序。

Lwp: SETB WR

Nop

CLR WR

Nop

SETB WR

Mov R6, #14H

写选通信号输入，由光传感器启动A/D转换。

DLAT: NOP

NOP

DJNZ R6, DLAT

MOV P1, #0FFH

SETB RD

NOP

CLR RD

NOP

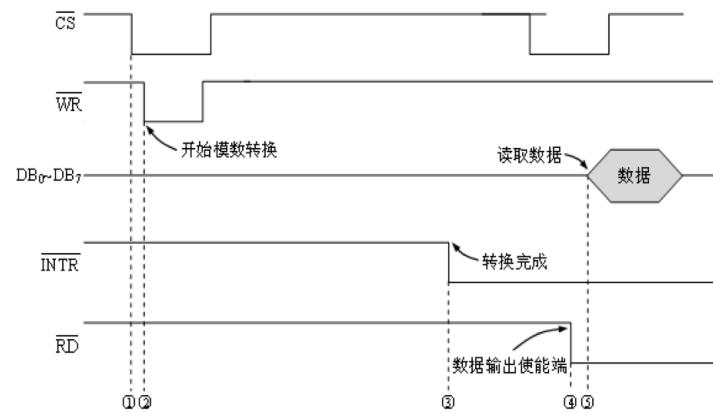
MOV A, P1

SETB RD

LCALL DELT

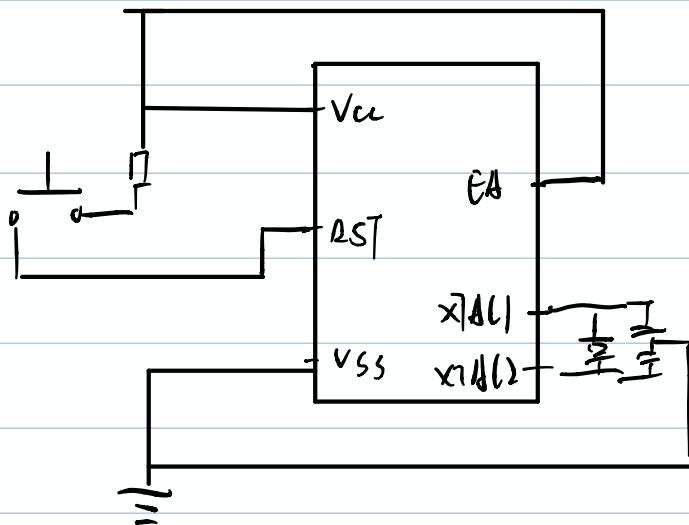
SJMP LOOP

loop延时



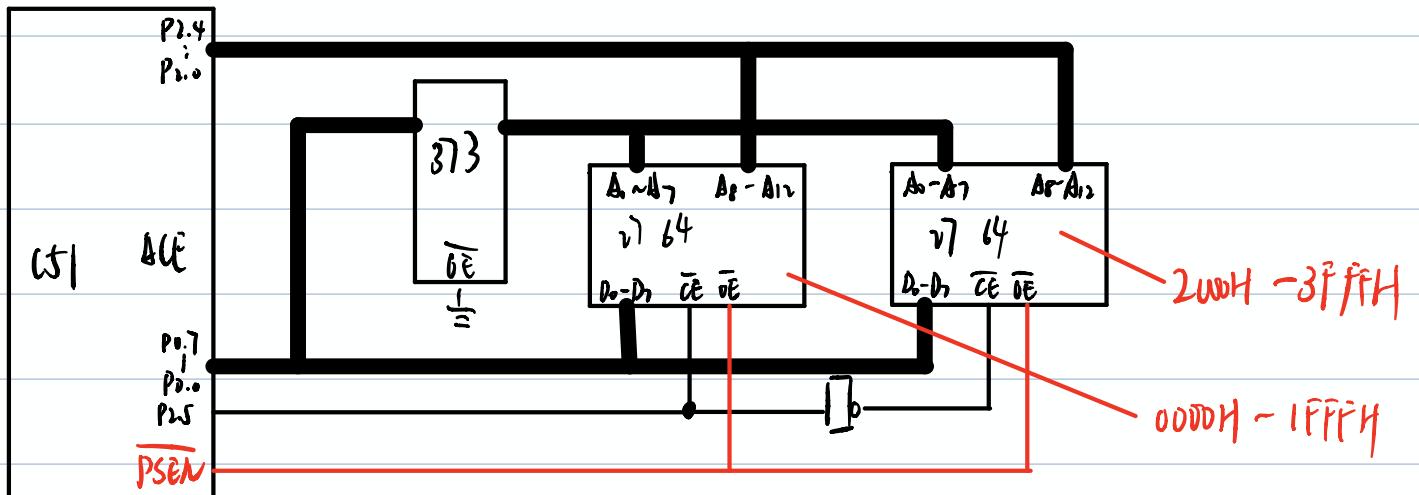
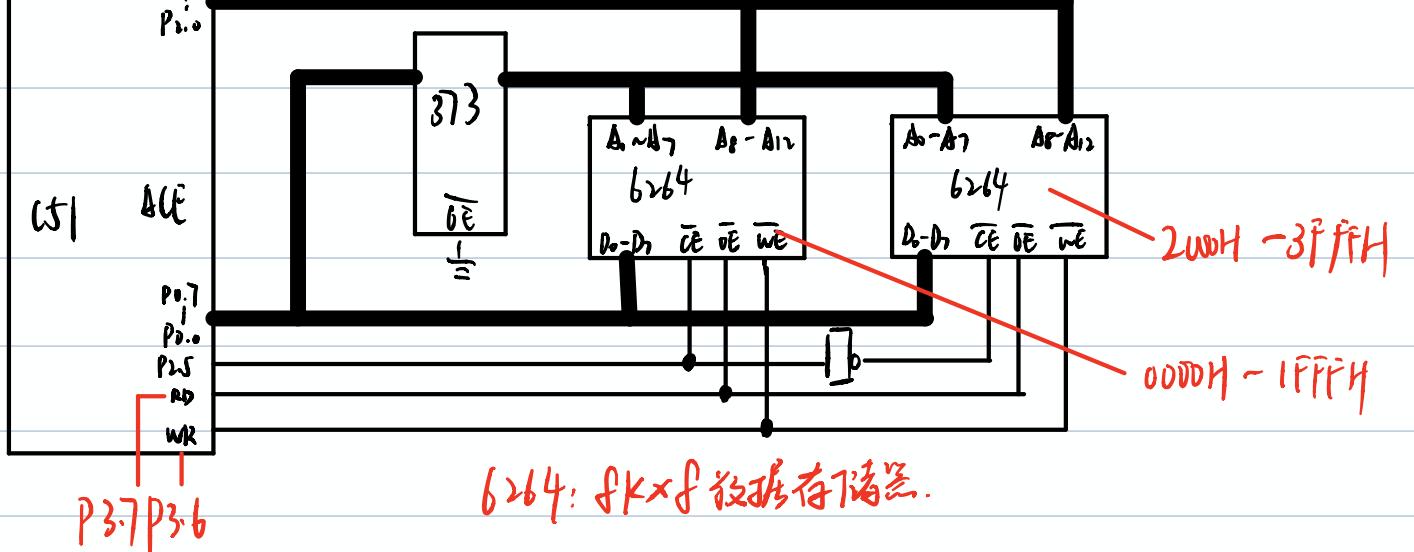
复位：ACJ1复位系统：

5V



关于软启程 / 数据存储器。

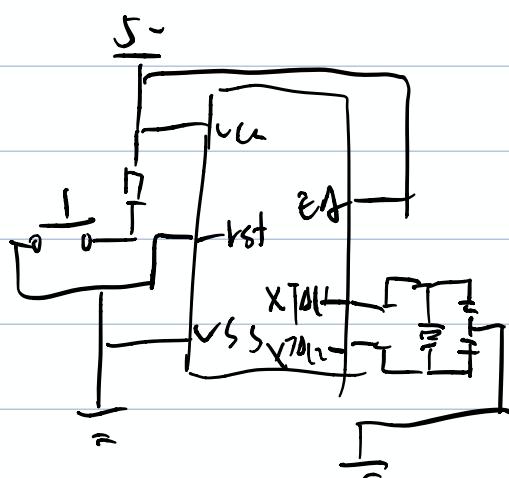
输出
P1.0 P1.1 P1.2 P1.3
AS0 AS1 AS2 AS3



复习题.

$$\text{机器周期} : \frac{12}{12\text{MHz}} = 1\text{MS}$$

指令周期 : 1.2 或 4 个机器周期.



0111	0100	A	1010
1110	1000	B	1011
		C	1100
		D	1101
		E	1110
		F	1111

$\delta \rightarrow 0H \quad 2H$
 SF 3H
 FE
 FC

1010 1010.

1110 0000
0000 1111

0001 0101

1011
B
1111

0110 1010

0100 0111

10110001

$A = B + C | H$
 $C = 0$

ADD : $\boxed{A} \leftarrow \boxed{B} + A$

{ 有进位 } Cy = 1

{ 3至4有进位 } AC = 1 变化相同.

~~6.7位有进位~~ $n=0$, 否则 $n=1$

A中有溢出 $|$, P=0, 否则 P=1

Mov DPLK, #20H

Mov A, 2H

Mov @DPLK, A

6MHz

02H 00H

ORG 0000H
 LJMP START
 LJMP 0000H
 START : MOV A, 0DH
 MOV TMOD, A
 MOV TH00

$$\frac{12}{6} = 2 \text{ μs 每次}$$

$$\frac{10 \times 10^3 \text{ μs}}{2} = 5 \text{ μs}$$

SETB T120
 SETB T120 EA

$T_{M0D} \rightarrow TH, TL \rightarrow ET, EA \rightarrow T12x$

计算延时时间：↓机器周期, $TP = 12/f_{osc}$.

$$TC = X \cdot TP$$

↑高字节 · ↑计数次数

$$X = TC / TP$$

计数初值: $2^n - X$. (看工作方式决定n). $= 5 \text{ μs}$

MOV TH00, #20H
 MOV TH1, ?
 MOV TL1, ?
 MOV PCON, #00H
 MOV SCON, #00H
 SETB TH0

RD BIT P3.6

WR BIT P3.7

SETB P2.7

SETB P0.7

WOP SETB WR

MOP

CLR

MOP

SETB WR

L CALL DELAY

MOV PI, OFFH

SETB RD

MOP

CLR RD

MOP

MOV VA, PI.

SETB RD

L CALL DELAY

SJMP

立即寻址

直接寻址

寄存器寻址

寄存器间接寻址

相对寻址

绝对

