

第二部分 COP8 系列单片机应用

目录

AN-521 双音多频 DTMF(Dual Tone Multiple Frequency)	(2-2)
AN-579 COP800 系列微控制器的微总线 ⁺ ™串行接口	(2-12)
AN-596 COP800 MathPak	(2-25)
AN-607 用 COP800 系列微控制器进行脉冲宽度调制的 A/D 转换技术	(2-65)
AN-662 基于 COP800 系列微控制器的自动安全/监督系统	(2-73)
AN-663 COP800 系列微控制器的音响效果	(2-81)
AN-666 用 3.58MHz 的晶体产生 DTMF	(2-104)
AN-673 利用 COP8 微控制器系列,使用 V/F 技术实现双向多路 LCD 驱动器和 低成本 A/D 转换器	(2-134)
AN-681 用 COP800 系列微控制器实现 PC ^R 鼠标器	(2-154)
AN-714 用 COP800 微控制器系列来控制直流(DC)步进电机	(2-182)
AN-734 具有 COP8 系列微控制器的 MF2 兼容键盘	(2-193)
AN-739 COP800 系列微控制器与 RS-232C 接口	(2-212)
AN-952 使用 COP800 系列微控制器实现低成本的 A/D 转换	(2-224)
AN-953 用 COP820CJ 实现 LCD 三路驱动器	(2-234)

AN-521 双音多频 DTMF(Dual Tone Multiple Frequency)

DTMF 的应用与数字电话有关,它提供了两个持续时间在 100ms 范围内可选择的输出频率(一个高频带、一个低频带)。在这节,给出了为 COP820C/840C 微控制器而编写的基准子程序,并详细地说明了这一程序。这个 DTMF 子程序占用了 110 个字节的 COP820C/840C 代码。它包括 78 个程序代码字节和 32 个 ROM 表字节。在 DTMF 子程序中,时序是建立在 20MHz 的 COP820C/840C 时钟基础上,且给出了 $1\mu\text{s}$ 的指令周期时间。

图 1 给出了与每个键有关的高频带或低频带频率的选择矩阵,它通过选择与矩阵行有关的四个低频带频率之一—唯一定位每个键,并使其选择与矩阵列有关的四个高频带频率之一联接。低频带频率为 697, 770, 852, 941Hz。高频带频率为 1209, 1336, 1447, 1633Hz。DTMF 程序假定:在累加器中,按照低位十六进制数位提供键译码。COP820C/840C DTMF 程序将分别在口 G 的输出引脚 G3 和 G2 上产生持续时间在 100ms 内的高频带频率和低频带频率。

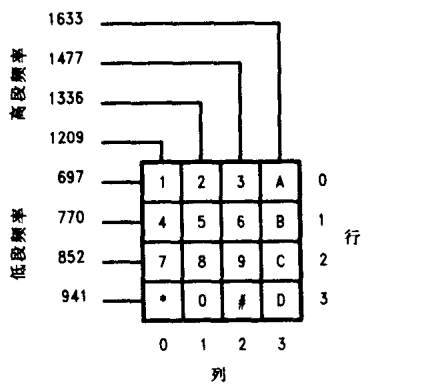


图 1 DTMF 键盘矩阵

每个 COP820C/840C 只包含一个定时器。可能出现的问题是必须产生三个不同的时间来满足应用的需要。这三个时间是两个所选择频率的周期和 100ms 的持续时间周期。显然,单个定时器可用来产生任何一个(有可能为两个)所需时间,同时,程序必须产生其他两个(或一个)时间。

DTMF 问题的解决办法在于对 8 个频率的每一个半个周期(四舍五入至微秒)去除 100ms 的持续时间,紧接着检查各自的高频带和低频带商数和

余数。这些除法的结果在表 1 中已详细地列出。低频带频率的商数范围为 139~188,而高频带商数的范围为 241~326。可以看到,只有低频带频率的商数能对应一个字节,这就使高频带频率要由 PWM(脉冲宽度调制)方式中运行的 16 位(2 字节)的 COP820C/840C 定时器来生成。

表 1 频率半周期、商和余数

	频率 Hz	半周期 (0.5P)	半周期 (μs)	100ms/0.5P	
				商数	余数
低频带频率	697	717.36	717	139	337
	770	649.35	649	154	54
	852	586.85	587	170	210
	941	531.35	531	188	172
高频带频率	1209	413.56	414 (256+158)	241	226
	1336	374.25	374 (256+118)	267	142
	1477	338.52	339 (256+83)	294	334
	1633	306.18	306 (256+50)	326	244

现在可看到,利用程序产生选择的低频带频率,同时记录 100ms 的持续时间就可以解决 DTMF 程序问题。而要做到这点,可以通过使用三个可编程寄存器计数器 R0, R2 和 R3,并用一个后备计数器 R1 重新加载计数器 R0,则可以实现。R0, R1 和 R3 代表与 4 个低频带频率相关的半周期, 100ms 的商和 100ms 的余数。产生所选低频带频率的工作原理首先是用从 ROM 表中取得的值加载三个计数器。经过所选择的低频带频率的半周期后,触发 G2 输出位。在半周期的计数过程中,商计数器减小。这一过程一直重复到商计数器数完为止,随后,程序转移到余数循环中。在余数循环中,余数计数器计数到 100ms 终止。当余数计数器数完后, G2 和 G3 复位,且 DTMF 子程序退出。在平衡所选择的低频带频率的半周期循环时,必须采取及时的保护措施,且 G2 输出位的触发(可用复位和置位指令来完成)也必须是准确的保持半周期时间完整性的平衡时间。局部小间隔循环(包括一个 DRSZ 指令,紧接着是由 JP 返回 DRSZ 的一个两字节, 6 个周期循环组成)嵌入在半周期和余数循环中。因此,ROM 表的半周期参

数和余数计数器大约只为期望值的六分之一。图 2 详细的时间平衡。
给出了半周期循环的程序和每个低频带频率循环的

程 序		字节/周期	条件周期		周期数	总周期数
	LD B, #PORTGD	2/3				
	LD X, #R1	2/3				
LUP1:	LD A[X-]	1/3			3	
	IFBIT 2,[B]	1/1			1	
	JP BYP1	1/3	3	1		
	X A,[X+]	1/3		3		
	SBIT 2,[B]	1/1		1		
	JP BYP2	1/3		3		
BYP1:	NOP	1/1	1			
	RBIT 2,[B]	1/1	1			
	X A,[X+]	1/3	3			
BYP2:	DRSZ R2	1/3 减 1			3	
	JP LUP2	1/3 Q 计数			3	
	JP FINI	1/3				
LUP2:	DRSZ R0	1/3 减 1		3	3	
	JP LUP2	1/3 F 计数		3	1	
	NOP	1/1			1	
	LD A,[X]	1/3			3	
	IFEQ A, #104	2/2			2	
	JP LUP1	1/3		1	3	31
	NOP	1/1		1		
	IFEQ A, #93	2/2		2		
BACK:	JP LUP1	1/3	1	3		35
	JP BACK	1/3	3			
			3			39

表 N 的频率×延时循环+总周期数=半周期
((114-1)×6)+39=717
((104-1)×6)+31=649
((93-1)×6)+35=587
((83-1)×6)+39=531

图 2 半周期循环时间平衡

DTMF 程序利用了两个 16 字节的 ROM 表。第一个 ROM 表包括输入 16 进制数到内部向量的转换表。16 进制数的编码与 16 进制数 ROM 表一起在表 I 中给出。图 1 给出的键盘矩阵的行和列位(RR, CC)分别代表低频带频率和高频带频率。它们被分别编码在 16 进制数的两个高位和两个低位。因此, 16 进制数的格式为 RRCC,同时,累加器中的输入字节由 0000RRCC 组成。在使用该值设置 16 进制数 ROM 转换表地址之前,程序把这个值变为 1101RRCC。

16 进制数 ROM 转换表中内部向量的组成格式为××00TT00,这里两个 T(定时器 Timer)位选择四个高频带频率中的一个。两个×选择四个低频带频率中的一个。对第二个 ROM 表,把内部向量转换成四个不同的输入,表 II 给出了这个转换过程。内部向量转换产生了一个定时器向量 1100TT00(T)和三个编程的计数器向量 R1,R2 和 R3。对应于 R1, R2,R3 的三个计数器向量的格式分别为 1100××11(F),1100××10(Q)和 1100××01(R)。随后,把由内部向量上产生的这四个向量用做第二个 ROM

表的输入。这四个向量(T 向量)之一是内存向量中 T 位的函数,其他三个向量(F,Q,R)为 X 位的函数。这样,只有一个参数与计时器的要求有关系(代表选择的高频带频率),同时,其他三个计数器需要

的三个参数(半周期、100ms 的商,100ms 的余数)与低频带频率和 100ms 的持续时间有关。表 IV 中给出了由 T、F、Q 和 R 向量访问的频率参数 ROM 转换表。

表 I 十六进制数字 ROM 转换表

	0	1	2	3
行	697 Hz	770 Hz	852 Hz	941 Hz
列	1209 Hz	1336 Hz	1477Hz	1633 Hz
地址	数据 (HEX)		键盘	
*			* 16 进制数字为 RRCC	
0xD0	000		1	其中 R = 行 #
0xD1	004		2	C = 列 #
0xD2	008		3	—— 例如:键 3 为行 #0,
0xD3	00C		A	列 #2,因此 16 进制数字
0xD4	040		4	是 0010 = 2
0xD5	044		5	RRCC
0xD6	048		6	
0xD7	04C		B	
0xD8	080		7	
0xD9	084		8	
0xDA	088		9	
0xDB	08C		C	
0xDC	0C0		*	
0xDD	0C4		0	
0xDE	0C8		#	
0xDF	0CC		D	

表 II 核心向量转换表

核心向量 -- XX00TT00 -----			
			*
			**

定时器向量	定时器	T	1100TT00
半周期向量	R1	F	1100XX11
商数向量	R2	Q	1100XX10
余数向量	R3	R	1100XX01

表 IV 频率参数 ROM 转换表

T -- 定时器	F -- 频率	Q -- 商数	R -- 余数
地址	数据 (DEC)	向量	
0xC0	158	T	

0xC1	53	R
0xC2	140	Q
0xC3	114	F
0xC4	118	T
0xC5	6	R
0xC6	155	Q
0xC7	104	F
0xC8	83	T
0xC9	32	R
0xCA	171	Q
0xCB	93	F
0xCC	50	T
0xCD	25	R
0xCE	189	Q
0xCF	83	F

概括地讲,输入 16 进制数字选择了第一个 ROM 表中的 16 个核心向量中的一个。然后把这个核心向量转换成四个其他向量(T,F,Q,R),并反过来使用这四个向量从第二个 ROM 表中选择四个参数。这四个参数用来加载定时器、相应的半周期、商及余数计数器。第一个 ROM 表(代表 16 进制数字矩阵表)从 ROM 存储单元 01D0 开始,可以任意设置,并且用 ADD A, #0D0 指令进行基准设置。第二个 ROM 表(代表频率参数表)必须从 ROM 存储单元 01C0(或 0XC0)开始设置以使程序最短,并且用 OR A, #0C3 指令进行 F 向量的基准设置,用 OR A, #0C0 指令进行 T 向量的基准设置。

与内部向量的两个 XX 有关的三个参数需要用 LAID 指令进行多级查表的功能。这点可由下面的 DTMF 程序中的部分代码实现:

```

LD      B, #R1
LD      X, #R4
X        A,[X]
LUP:    LD      A,[X]
        LAID
X        A,[B+]
DRSZ    R4
IFBNE   #4
JP      LUP

```

这段程序代码把 F 频率向量加载到 R4 中,并且每次完成循环时都使向量减 1。R4 向量的连续循环递减把 F 向量转变为 Q 向量;接着把 Q 向量变为 R 向量。R4 向量通过 LAID 指令访问 ROM 表。X 指针参照 R4 向量,B 指针在用来储存三个所选择的 ROM 表参数(每次循环存储一个参数)之后,每次通过循环都加 1。这三个参数储存在连续的 RAM 存储单元 R1,R2 和 R3 中。一旦三个所选择的 ROM 表参数已经访问或存储,IFBNE 检测指令就跳出循环。

定时器初始计数为 15,因此,定时器首次下溢和 G3 输出位触发(所选择定时器 PWM 方式和 G3 触发输出)与 G2 输出位的第一次触发同时发生。高频带半周期的频率范围为 306~414,因此,在第二个 ROM 表的定时器中储存了这些值减 256 的值,然后把从这个频率 ROM 表中所选择的值存储在定时器自动重装寄存器的下半字节,同时在上半字节存储 1。计时器被选为 PWM 输出方式,并且从指令 LD[B], #0B0 开始计时,这里 B 指针在存储器存储单元 0EE 上选择 CNTRL 寄存器。

COP820C/840C 的 DTMF 子程序使用了 110 个字节的代码,包括 78 字节程序代码和 32 字节 ROM 表。向为每个 16 进制数字输入调用 DTMF 子程序的程序提供了 DTMF 子程序列表。

国家半导体公司

COP800 交叉汇编程序,REV:B,20 JAN 87

```

1          ;COP820/840C 的 DTMF 程序
2          ;
3          ;DTMF—双音多频
4          ;
5          ;程序名:DTMF.MAC
6          ;
7          .TITLE DTMF
8          .CHIP840
9          ;    * * * * * DTMF 子程序包含 110 个字节 * * * * *
10         ;    * * * * * DTMF 在 100ms 后定时结束 * * * * *
11         ;    * * 从第一个 G2/G3 输出的触发开始 * *
12         ;    * * * 基于 20MHz 的 COP820C/840C 时钟 * * *
13         ;
14         ;G 口用于两种输出
15         ;    — 高频带(HB)频率输出在 G3
16         ;    — 低频带(LB)频率输出在 G2
17         ;
18         ;定时器计数结束
19         ;    — HB 频率
20         ;
21         ;程序计数结束
22         ;    — LB 频率
23         ;    — 100ms 被 LB 半周期商除
24         ;    — 100ms 被 LB 半周期余数除
25         ;
26         ;16 个 16 进制数字矩阵向量格式为 1101RRCC,
27         ; 其中—RR 为行选择(LB 频率)
28         ;    CC 为列选择(HB 频率)
29         ;
30         ;矩阵选择表中的 16 个核心(CORE)向量格式为 XX00TT00,
31         ; 其中—TT 是 HB 选择
32         ;    XX 是 LB 选择
33         ;
34         ;由核心向量产生的 FREQ 参数表的
35         ; 频率向量(HB 和 LB)
36         ;
37         ;对于定时器计数,HB 频率向量(4)以 00 结束,
38         ; 其中向量格式为 1100TT00
39         ;
40         ;LB 频率向量以下述结束:

```

```

41          ; 11 用于半周期循环计数,
42          ;      其中向量格式为 1100XX11
43          ; 10 用于 100ms 被半周期商数除时,
44          ;      其中向量格式为 1100XX10
45          ; 01 用于 100ms 被半周期余数除时,
46          ;      其中向量格式为 1100XX01
47          ;
48          ;16 进制 01D * 形式的 16 进制数字矩阵表(可选,
49          ; 依赖于 'ADD A, #0D0' INST. 立即值)
50          ;
51          ;16 进制 01C * 形式的 FREQ 参数表(必需)
52          .FORM
53          ;
54          ;MAGIC: 核心向量
55          ;      XX00TT00
56          ;
57          ; 定时器      T      TT00
58          ;  R1      F      XX11
59          ;  R2      Q      XX10
60          ;  R3      R      XX01
61          ;
62          ;说明
63          00D0      PORTLD=0D0      ;PORTL 数据寄存器
64          00D1      PORTLC=0D1      ;PORTL 配置寄存器
65          00D4      PORTGD=0D4      ;PORTG 数据寄存器
66          00D5      PORTGC=0D5      ;PORTG 配置寄存器
67          00DC      PORTD=0DC      ;PORTD 寄存器
68          00EA      TIMERLO=0EA      ;定时器低计数器
69          00EE      CNTRL=0EE      ;控制寄存器
70          00EF      PSW=0EF      ;过程状态字
71          00F0      R0=0F0      ;LB 频率循环计数器
72          00F1      R1=0F1      ;LB 频率循环计数
73          00F2      R2=0F2      ;LB 频率 Q 计数
74          00F3      R3=0F3      ;LB 频率 R 计数
75          00F4      R4=0F4      ;LB 频率表向量
76          ;
77 0000 DD2F      START:      LD      SP, #02F      ;16 进制数字矩阵
78 0002 BCD1FF      LD      PORTLC, #0FF      ;1 2 3 A
79 0005 BCD080      LD      PORTLD, #080      ;4 5 6 B
80 0008 DEDC      LD      B, #PORTD      ;7 8 9 C
81 000A 9E00      LD      [B], #0      ; * 0 # D
82 000C AE      LOOP:      LD      A, [B]      ;DTMF 测试循环
83 000D 3160      JSR      DTMF      ;到 16 进制矩阵

```

84 000F DEDC		LD	B, #PORTD	;到子程序
85 0011 AE		LD	A, [B]	;输出到 PORTD
86 0012 9405		ADD	A, #5	;当触发时执行
87 0014 A6		X	A, [B]	;每一次调用 DTMF
88 0015 6C		RBIT	4, [B]	;子例程时 PORTL
89 0016 9DD0		LD	A, PORTLD	;可同时输出
90 0018 A1		SC		;输出序号
91 0019 B0		RRC	A	;1,5,9,D,4,8,#,A
92 001A 9CD0		X	A, PORTLD	;7,0,3,B,* ,2,6,C
93 001C EF		JP	LOOP	
94	;			
96	;			
97 0160			, =0160	
98	;			
99 0160 DED5	DTMF:	LD	B, #PORTGC	
100 0162 9B3F		LD	[B-], #03F	
101 0164 6B		RBIT	3, [B]	;任选
102 0165 6A		RBIT	2, [B]	;任选
103	;			
104 0166 94D0		ADD	A, #0D0	
105 0168 A4		LAIID		;数字矩阵表
106	;			
107 0169 5F		LD	B, #0	
108 016A A6		X	A, [B]	
109 016B AE		LD	A, [B]	
110 017B 65		SWAP	A	
111 016C 97C3		OR	A, #0C3	
112 016E DEF1		LD	B, #R1	
113 0170 DCF4		LD	X, #R4	
114 0172 B6		X	A, [X]	
115 0173 BE	LUP:	LD	A, [X]	
116 0174 A4		LAIID		;LB 频率表
117 0175 A2		X	A, [B+]	; (3 参数)
118 0176 C4		DRSZ	R4	
119 0177 44		IFBNE	#4	
120 0178 FA		JP	LUP	
121	;			
122 0179 5F		LD	B, #0	
123 017A AE		LD	A, [B]	
124 017C 97C0		OR	A, #0C0	
125 017E A4		LAIID		;HB 频率表
126 017F DEEA		LD	B, #TIMERLO	; (1 参数)
127 0181 9A0F		LD	[B+], #15	

128 0183 9A00		LD	[B+], # 0	
129 0185 A2		X	A, [B+]	
130 0186 9A01		LD	[B+], # 1	
131 0188 9EB0		LD	[B], # 0B0	;启动定时器 PWM
132	;			
133 018A DED4		LD	B, # PORTGD	
134 018C DCF1		LD	X, # R1	
135	;			
136 018E BB	LUP1:	LD	A, [X-]	
137 018F 72		IFBIT	2, [B]	;测试 LB 输出
138 0190 03		JP	BYP1	
139 0191 B2		X	A, [X+]	
140 0192 7A		SBIT	2, [B]	;设置 LB 输出
141 0193 03		JP	BYP2	
142 0194 B8	BYP1:	NOP		
143 0195 6A		RBIT	2, [B]	;LB 输出复位
144 0196 B2		X	A, [X+]	
145 0197 C2	BYP2:	DRSZ	R2	;DECR. QUOT. COUNT
146 0198 01		JP	LUP2	
147 0199 0C		JP	FINI	;Q 计数完成
148	;			
149 019A C0	LUP2:	DRSZ	R0	;DECR. F COUNT
150 019B FE		JP	LUP2	;LB(半周期)
151	;			
152 019C B8		NOP		; * * * * *
153 019D BE		LD	A, [X]	;平衡
154 019E 9268		IFEQ	A, # 104	;LB 频率
155 01A0 ED		JP	LUP1	;的半周期
156	;			;驻留
157 01A1 B8		NOP		;延迟
158 01A2 925D		IFEQ	A, # 93	; (对每一个
159 01A4 E9	BACK:	JP	LUP1	;4LB 频率)
160 01A5 FE		JP	BACK	; * * * * *
161	;			
162 01A6 C3	FINI:	DRSZ	R3	;DECR. REM. COUNT
163 01A7 FE		JP	FINI	;R 计数未完成
164	;			
165 01A8 BDEE6C		RBIT	4, CNTRL	;停止定时器
166 01AB 6B		RBIT	3, [B]	;清 HB 输出
167 01AC 6A		RBIT	2, [B]	;清 LB 输出
168	;			
169 01AD 8E		RET		
170	;			

```

171          . FORM
172          ;
173          ;频率和 100ms 参数表
174      01C0          . =01C0
175          ;
176 01C0 9E          . BYTE      158          ;T
177 01C1 35          . BYTE      53          ;R
178 01C2 8C          . BYTE      140         ;Q
179 01C3 72          . BYTE      114         ;F
180 01C4 76          . BYTE      118         ;T
181 01C5 06          . BYTE      6           ;R
182 01C6 9B          . BYTE      155         ;Q
183 01C7 68          . BYTE      104         ;F
184 01C8 53          . BYTE      83          ;T
185 01C9 20          . BYTE      32          ;R
186 01CA AB          . BYTE      171         ;Q
187 01CB 5D          . BYTE      93          ;F
188 01CC 32          . BYTE      50          ;T
189 01CD 19          . BYTE      25          ;R
190 01CE BD          . BYTE      189         ;Q
191 01CF 53          . BYTE      83          ;F
192          ;
193          ;数字矩阵表
194      01D0          =. 01D0
195          ;
196 01D0 00          . BYTE      000          ;1      0      0
197 01D1 04          . BYTE      004          ;2      0      1
198 01D2 08          . BYTE      008          ;3      0      2
199 01D3 0C          . BYTE      00C          ;A      0      3
200 01D4 40          . BYTE      040          ;4      1      0
201 01D5 44          . BYTE      044          ;5      1      1
202 01D6 48          . BYTE      048          ;6      1      2
203 01D7 4C          . BYTE      04C          ;B      1      3
204 01D8 80          . BYTE      080          ;7      2      0
205 01D9 84          . BYTE      084          ;8      2      1
206 01DA 88          . BYTE      088          ;9      2      2
207 01DB 8C          . BYTE      08C          ;C      2      3
208 01DC C0          . BYTE      0C0          ; *     3      0
209 01DD C4          . BYTE      0C4          ;0      3      1
210 01DE C8          . BYTE      0C8          ;#      3      2
211 01DF CC          . BYTE      0CC          ;D      3      3
212
213          . END

```

符号表 (SYMBOL TABLE)

B	00FE	BACK	01A4	BYP1	0194	BYP2	0197
CNTRL	00EE	DTMF	0160	FINI	01A6	LOOP	000C
LUP	0174	LUP1	018E	LUP2	019A	PORTD	00DC
PORTGC	00D5	PORTGD	00D4	PORTLC	00D1	PORTLD	00D0
PSW	00EF *	R0	00F0	R1	00F1	R2	00F2
R3	00F3	R4	00F4	SP	00FD	START	0000 *
TIMERL	00EA	X	00FC				

宏表 (MACRO TABLE)

NO WARNING LINES
 NO ERROR LINES
 139 ROM BYTES USED

SOURCE CHECKSUM=99A7

OBJECT CHECKSUM=003E1

INPUT FILE C:\DTMF.MAC

LISTING FILE C:\DTMF.PRN

OBJECT FILE C:\DTMF.LM

该应用说明中列出的代码可求助于 Dial-A-Helper。

Dial-A-Helper 是由微控制器应用部门提供的服务。系统提供了对自动信息系统和检索系统的访问,该系统可通过拨码电话线进行全天访问。系统包括一个与微控制器应用部门交换信息的 MESSAGE SECTION(电子邮件)和一个可用于检索与 NSC 微控制器有关的应用数据的 FILE SECTION。访问 Dial-A-Helper 系统的最小需求是一个 300 或 1200 波特率的调制解调器及一部电话。

若用户有一台带有通信软件包的 PC 机,则 FILE SECTION 中的文件可拷入到磁盘以备将来使用。Dial-A-Helper 的电话是:

Modem: (408)739-1162

Voice: (408)721-5582

需要其它信息可以与工厂联系

AN-579 COP800 系列微控制器的微总线⁺™ 串行接口

介绍

国家半导体公司的多功能、成本低廉的微控制器 COP800 系列,使用了 8 位单片内核结构,采用 M²CMOS 制造工艺。这些高性能的微控制器为高效系统提供了一个多功能指令系统和高级功能。

COP800 系列微控制器以串行通信微总线⁺™ 方式特征。微总线⁺™ 是微总线™ 同步串行通信方式的增强型,它最初是在 COP400 系列微控制器上实现的。在 COP800 系列微控制器上的微总线接口容易进行 I/O 扩展,并且与许多 COPS 系列微控制器外设(A/D 转换器,EEPROM,显示器,驱动器等)进行接口,且与其它支持微总线或串行接口的 SPI⁺ 方式的微控制器进行接口。

微总线⁺™ 的定义

微总线⁺™ 是一个多用途的三线,SI(串行输入)、SO(串行输出)和 SK(串行时钟)的双向串行通信系统,其中 COP800 或者是提供移位时钟(SK)的主控制器或者是接收外部时钟(SK)的从属控制器。图 1 给出了 COP800 微总线⁺™ 系统框图。微总线⁺™ 串行接口采用 8 位存储器映射微总线⁺™ 串行移位寄存器 SIOR,其中 SIOR 由 SK 信号同步。如其名所述,SIOR 寄存器用作串行传输的移位寄存器。COP800 微控制器的串行输入引线 SI 是移位寄存器的输入。移位寄存器的输出 SO 是外部装置的串行输出。SK 是串行同步时钟。用 SK 时钟同步外部设备数据的

输入和输出。SO、SI 和 SK 分别映射为 8 位双向 G 口的引脚 4、5、6 的复用功能。

微总线⁺™ 的操作

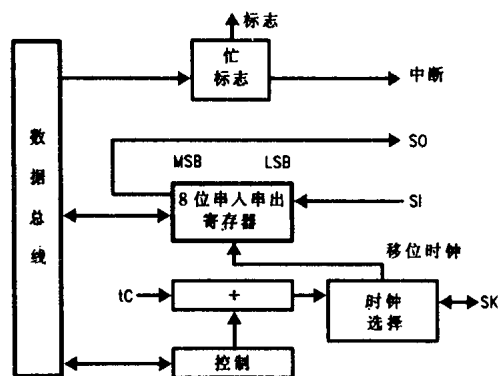
在微总线⁺™ 串行接口中,SI 引脚上的输入数据的移位顺序是首先把高位移入 8 位 SIOR 移位寄存器的最低有效位(LSB),输出数据的移位顺序是首先把移位寄存器中的最高有效位移出到 SO 引脚。由 SK 时钟信号下降沿同步 SIOR 寄存器。SI 引脚上的输入数据在 SK 时钟上升沿移到 SIOR 寄存器的 LSB。在 SK 时钟信号的下降沿,SIOR 寄存器的 MSB 位移出到 SO 引脚。

在微总线⁺™ 操作的主控方式,COP800 内部产生 SK 时钟信号。在从属方式,由外部装置产生 SK 时钟,并输入到 COP800 中。

在 CNTRL 寄存器中的 MSEL(微总线选择)标志用来启动微总线⁺™ 操作。设置 MSEL 标志使微总线⁺™ 接口的选通信号通过 G 端口。信号 SO、SK 和 SI 分别使用了 G 口的引脚 G4、G5、G6。应该注意,G 口配置寄存器必须根据微总线⁺™ 的要求进行合理的配置。表 1 说明了 G 口配置。在微总线⁺™ 操作的主控方式中,要把 G4 和 G5 选作 SK 和 SO 信号的输出,反过来,在从属操作方式中,要把 G5 配置为外部 SK 时钟的输入。SI 信号是 G6 引脚的专用输入,因而不需要进一步的设置。

表 1 G 口配置

G4(SO) 配置位	G4(SK) 配置位	G4 功能	G5 功能	操作方式
1	1	SO	内部 SK	微总线主控方式
0	1	三态	内部 SK	微总线主控方式
1	0	SO	外部 SK	微总线从属方式
0	0	三态	外部 SK	微总线从属方式



TL/DD/10252-1

图 1 微总线⁺™ 功能图

在微总线⁺™ 主控方式操作中,CNTRL 寄存器的 SL1 和 SL0 位(COP820C 和 COP840C 中的 S1 和 S0)用来选择时钟发生的时钟分频因子(2,4 或 8),表 2 给出了 CNTRL 寄存器的这些位以及 CNTRL 寄存器的时钟选择表。当微总线⁺™ BUSY 标志为低

电平时,与主控方式时钟分频因子有关的计数器被清零。时钟分频因子与指令周期频率有关。例如,如果 COP800 以 1MHz 的内部时钟运行时,则 SK 时钟速率将为 500kHz、250kHz 或 125kHz,对应的 SL1 和 SL0 分别为 00、01 和 10(或 11)。

表 1

CNTRL 寄存器(地址 X'00EE)

定时器 1(T1)和微总线控制寄存器包括下列位:

SL1 和 SL0	选择微总线时钟分频因子(00=2,01=4,10=8)
IEDG	外部中断沿极性选择(0=上升沿,1=下降沿)
MSEL	选择 G5 和 G4 分别作为微总线信号 SK 和 SO。
T1C0	在定时方式 1 和 2 中,定时器 T1 起始/停止控制。 在定时方式 3 中,定时器 T1 下溢中断挂起标志。
T1C1	定时器 T1 方式控制位。
T1C2	定时器 T1 方式控制位。
T1C3	定时器 T1 方式控制位。

T1C3	T1C2	T1C1	T1C0	MSEL	IEDG	SL1	SL0
位 7							位 0

SL1	SL0	SK
0	0	$2 \times t_c$
0	1	$4 \times t_c$
1	x	$8 \times t_c$

t_c 为指令周期时钟

微总线+主控方式操作

在微总线+主控方式中,PSW(处理器状态字)的 BUSY 标志用来控制微总线+8 位移位寄存器的移位。设置 BUSY 标志可使 SIOR 寄存器在移位寄存器的高位端上,从 SO 上移出 8 位数据;同时,一个新的 8 位数据从 SI 中输入到 SIOR 寄存器的低位端上。在 8 位数据移位之后,BUSY 标志会自动复位(图 2)。当 BUSY 标志变为低电平,表示 8 位移位结束时,COP88XX 系列微控制器提供了矢量可屏蔽中断。在 SK 时钟的上升沿,来自 SI 引脚的输入数据同步输入到 SIOR 寄存器,而在 SK 时钟的下降沿上,SIOR 寄存器的 MSB 被移进 SO 引脚。用户可以通过软件复位 BUSY 位,进行小于 8 位的数据移

位。但是,要保证软件 BUSY 复位,只能在 SK 时钟低电平时进行,这是为了避免狭窄的 SK 终端时钟。

微总线+从属方式操作

在微总线+从属方式中,SK 时钟是由外部源产生的。在 CNTRL 寄存器中设置 MSEL 位使 G 口具有 SO 和 SK 功能。通过复位和设置 G 配置寄存器中的适当位,把 SK 引脚配置为输入,把 SO 引脚配置为输出。一旦进入从属方式,用户必须立即设置 BUSY 标志。在 8 个时钟脉冲之后,必须清除 BUSY 标志,并可重复该过程。不管怎样,在从属方式中,如果复位 BUSY 标志,COP888 系列微控制器就不会移位数据。而如果 SK 时钟有效,不管 BUSY 标志怎样,COP820C 和 COP840C 都将继续移位数据。

微总线+交替 SK 方式

COP888XX 系列微控制器也允许另外的交替 SK 相位操作。在正常操作方式中,数据是在 SK 时钟的上升沿移入,在下降沿上移出(图 2)。SIOR 寄存器是在每个 SK 时钟的下降沿上进行移位。在交替 SK 相位操作方式中,数据是在 SK 时钟的下降沿上移入,在上升沿上移出的(图 3)。

控制标志 SKSEL 允许选择正常的 SK 时钟或者选择交替的 SK 时钟,复位 SKSEL 选择正常的 SK 时钟,置位 SKSEL 选择交替的微总线+逻辑 SK 时钟,SKSEL 标志映射到 G6 的配置位。在加电后,复位 SKSEL 标志,选择正常的 SK 时钟信号。交替方式方便了与正常 SK 时钟操作不兼容的外设之间的串行数据传输的微总线+协议的使用。即:在 SK 时钟的下降沿上移出数据,在 SK 时钟的上升沿上移入数据。

微总线+采样协议

这一部分给出了使用 COP888CL 和 COP840C 的微总线+采样协议。它阐明了采样协议的从属方式操作过程,并提供了协议的时序说明。

1. 设置 CNTRL 寄存器中的 MSEL 位来启动微总线,配置 G0($\overline{CS}$)和 G5(SK)为输入,G4(SO)为输出,G6(SI)一直为输入。
2. 主机的片选线($\overline{CS}$)连接到从属装置的 G0, $\overline{CS}$ 引线的有效低电平使从属器件产生中断。
3. 随着 $\overline{CS}$ 引线上的从高电平到低电平的转变,在时间“T”之前,在微总线上不传输数据(参

照图 4)。

4. 通过开通 SK 时钟, 主控装置初始化微总线上的数据传输。
5. 在主控器件和从属器件之间进行一系列的

数据传输。

6. 主控器件使 $\overline{\text{CS}}$ 引线变为高电平来结束微总线操作, 从属器件返回正常操作方式。

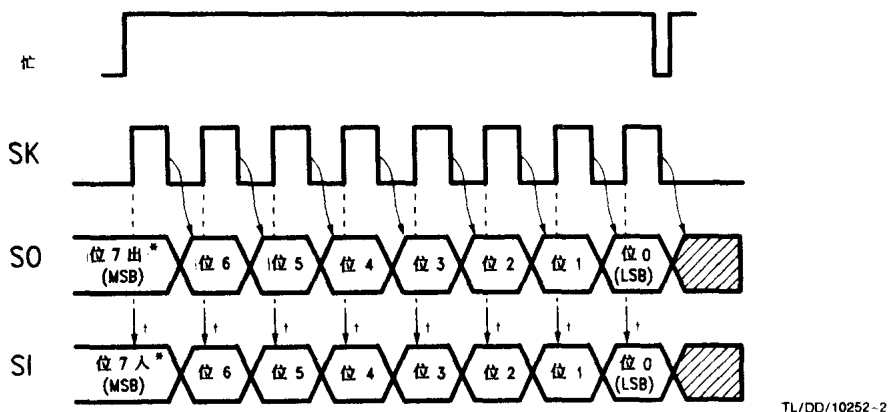


图 2 微总线⁻时序

* 在加载转换装置的 SIOR 寄存器后, 这个位立即有效。

↑ 箭头表示 SI 的采样点。

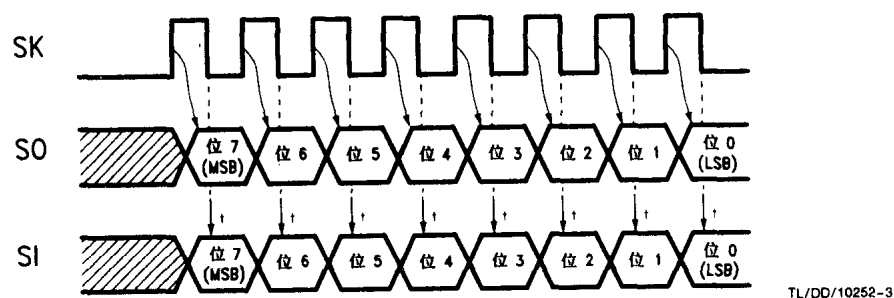


图 3 交替相位 SK 时钟时序

↑ 箭头表示 SI 的采样点。

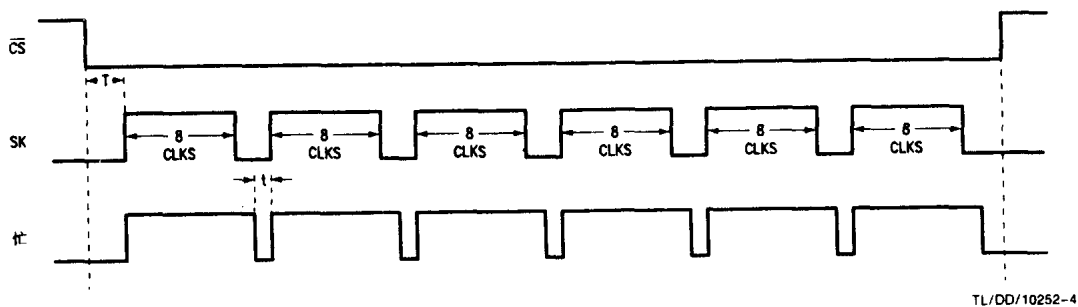


图 4 微总线⁻采样协议时序图

从属方式操作过程

1. 设置 CNTRL 寄存器中的 MSEL 位来允许微总线,配置 G0($\overline{CS}$)和 G5(SK)为输入,G4(SO)为输出,G6(SI)一般为输入。
2. 直到由 $\overline{CS}$ 变为低电平产生中断之前,一直为正常操作方式。
3. 设置 BUSY 标志,用 SO 上输出的数据加载 SIO 寄存器。(在移位寄存器的高位端移位寄存器从 SO 中移位 8 位数据。同时,把来自 SI 中的新的 8 位数据加载到移位寄存器的低位端。)
4. 等待 BUSY 标志复位(在 8 位数据移位之后,BUSY 标志将自动复位)。
5. 如果正在读入数据,用户应该保存 SIOR 寄存器的内容。
6. 完成预先设置的数据传输。
7. 重复第 3 步到第 6 步。用户必须按协议规定时间“t”内实现第 3 步到第 6 步。

COP888 和 COP820/COP840 间的差异

在某些方面,COP888 系列微总线⁺的特征与 COP820/840 有些不同。在由 BUSY 标志变为低电平表示的微总线⁺操作完成后,可配置 COP888 系列来中断处理器。COP888 系列支持矢量中断,为每个中断源保留了两字节的内存空间。用户可做任何所需的现场切换,然后编一条 VIS(中断向量选择)指令,以转移到在 VIS 指令期间允许和挂起的最高优先级中断的中断服务程序处。由用户选择不同的中断服务程序地址,并把它们存储在 ROM 中由 0yE0 开始的表中,这里“y”取决于 VIS 指令位置的 256 个字节(0y00 到 0yFF)。微总线⁺中断的地址为 0yF2-0yF3。

其次,COP888 系列支持微总线⁺操作的交替 SK 相位方式。这个特征便于与正常 SK 时钟方式不兼容的外设之间的微总线⁺串行数据传输协议的使用。即:在 SK 时钟的下降沿移出数据,在 SK 时钟的上升沿移入数据。

接口考虑

为保持使用微总线⁺进行数据交换的完整性,

必须考虑如下两个方面:

1. 串行数据交换时序。
2. 扇出/扇入的要求。

从理论上讲,无数个设备可以访问同一个接口,并能及时、唯一地逐次允许。实际上,可访问同一串行接口的实际设备数量取决于下面的条件:系统数据传输速度,系统电源要求,在 SK 上和 SO 输出上的负载电容,所接口的逻辑系列和分离元件的扇入要求。

硬件接口

为了使正确的数据传输出现在输出中,需要在预定的时间内进行高电平和低电平的转换。数据转换严格要求同步,并且时序与微总线⁺系统时钟(SK)有关。例如,如果在时钟下降沿上 COPS 控制器输出一个值,且在时钟的上升沿上由外部设备锁存,则下列关系必须满足:

$$t_{\text{DELAY}} + t_{\text{SETUP}} \leq t_{\text{CK}}$$

其中 t_{CK} 为从数据输出开始到转换到锁存在外设中的数据的时间; t_{SETUP} 为外部设备的设置时间,这里数据必须有效,且 t_{DELAY} 为从输出到读有效数据的时间。 t_{CK} 与由 COPS 控制器的 SK 引脚提供的系统时钟有关,并随 COPS 指令周期时间的增加而增加。

当与微总线⁺接口时,除了时序要求外,系统供电和扇入/扇出的要求也必须考虑。为了在同一个微总线⁺上驱动多个外设,控制器的输出驱动器需要源和吸收与其连接的所有输入的总的最大漏电流,并保持信号电平在有效逻辑电平“1”和“0”输入电压电平内。因此,若不同类型的元件与同一个串行接口连接时,控制器的输出驱动器必须满足各个外设的所有输入要求,同样,当具有三态输出的元件连接到 SI 输入时,必须满足控制器的最小有效输入电平和所有输出的最大三态漏电流。

因此,对那些具有不兼容输入电平或无源电流/吸收电流要求的元件,为了提供电平移位或驱动,就必须接外部上拉电阻和缓冲器。

表 III

特性		元件编号							
		DS890XX	MM545X	COP470	COP472	ADC83X (COP430)	COP498 /499	COP452L	NMC9306 (COP494)
总特性									
元件功能		AM/PM PLL	LED 显示 驱动器	VF 显示 驱动器	LCD 显示 驱动器	A/D	RAM 定时器	频率 发生器	E ² PROM
工艺		ECL	NMOS	PMOS	CMOS	CMOS	CMOS	NMOS	NMOS
V _{cc} 范围		4.75V~ 5.25V	4.5V~ 11V	-9.5V~ 4.5V	3.0V~ 5.5V	4.5V~ 0.3V	2.4V~ 5.5V	4.5V~ 6.3V	4.5V~ 5.5V
引脚数		20	40	20	20	8/14/20	14/8	14	14
硬件接口									
最小 V _{IH} / 最大 V _{IL}		2.1V/ 0.7V	2.2V/ 0.8V	-1.5V/ -4.0V	0.7V _{cc} / 0.8V	2.0V/ 0.8V	0.8V _{cc} / 0.4V _{cc}	2.0V/ 0.8V	2.0V/ 0.8V
SK 时钟范围		0-625kHz	0-500kHz	0-250kHz	4-250kHz	10-200kHz	4-250kHz	25-250kHz	0-250kHz
写数 据 DI	设置最 小时间	0.3μs	0.3μs	1.0μs	1.0μs	0.2μs	0.4μs	800ns	0.4μs
	保持最 小时间	0.8μs	(注 3)	50ns	100ns (注 3)	0.2μs	0.4μs	1.4μs	0.4μs
读数据 程序延迟		(注 4)	(注 3)	(注 3)	(注 3)	(注 3)	0.2μs (注 2)	1μs (注 2)	2.0μs
片允 许	设置 时间	0.275μs	0.4μs	1.0μs 最小	1μs (注 1)	0.2μs	0.2μs (注 1)	(注 3)	0.2μs
	保持 时间	0.300μs	(注 3)	1.0μs 最小	1μs (注 2)	0.2μs	0 (注 2)	(注 3)	0
最大 频率 范围	AM	8MHz	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)
	FM	120MHz	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)	(注 3)
最大振荡 频率		(注 3)	(注 3)	250kHz	(注 3)	(注 3)	2.1MHz(-21) 32kHz(-15)	256-2100kHz(-4) 64-525kHz(-2)	(注 3)
软件									
串行 I/O 协议		11D1 ~D20	1D1 ~D35	每次 8 位	b1 -b40	1xxx	1yyxxD6~D0 起始位	1yxxxx	1AA-DD
指令/地址字		无	无	无	无	(注 4)	(注 4)	(注 4)	(注 4)

注 1: 以 SK 的上升沿为基准

注 2: 以 SK 的下降沿为基准

注 3: 未定义

注 4: 参考不同的操作方式的数据手册

典型应用

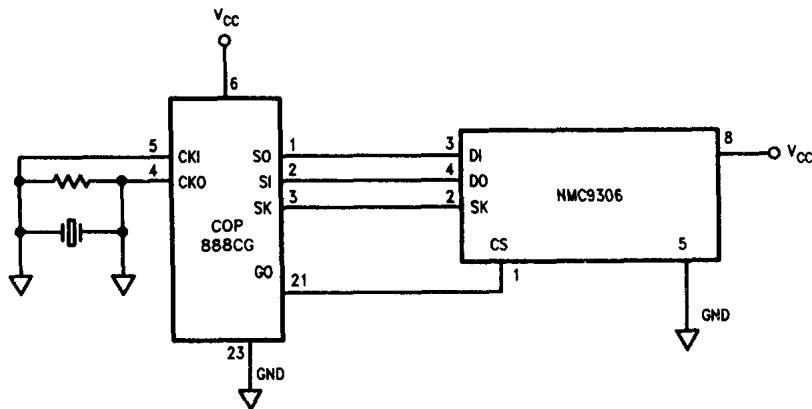
现有元件的整个系列直接与微总线* 协议兼容,这就允许与 COP800 微控制器系列直接接口。表 III 概括了现存元件及其功能和技术要求。

NMC9306-COP888CG 接口

图 5 给出了 NMC9306 (COP494), 256 位 E²PROM 与 COP888CG 接口的引脚连接图,有关 NMC9306 接口要求应注意以下几点:

- 1. SK 时钟的频率范围应为 0kHz~250kHz。

2. 擦除/写指令之后的 $\overline{CS}$ 低电平时间必须最大不超过 300ms, 应该设置为典型值或者最小值 10ms。
3. 当执行任何指令时, 由在 $\overline{CS}$ 启动("0"到"1")之后的"0"到"1"转换设置 DI 上的起始位。
4. 读方式中, 在指令和数据链之后, 当数据正在输出时即对随后的 17 位或时钟, DI 可以"忽略"。在指令和数据输入后, 这点对其它指令也成立。
5. 数据输出链从空闲位 0 开始, 通过禁止芯片终止数据链。16 位数据后的额外 SK 周期是多余的。如果在所有 16 位数据输出之后, $\overline{CS}$ 保持允许状态, 则在另一个 $\overline{CS}$ LO 到 HI 转换开始一个新的指令周期之前, DO 将一直输出 DI 的状态。
6. 读周期之后, 在另一个指令周期开始之前, $\overline{CS}$ 必须为低电平, 并保持一个 SK 时钟周期。



TL/DD/10252-5

图 5 NMC9306-COP888CG 接口

指令系统

命令	起始位	操作码	地址	注释
READ	1	0000	A3A2A1A0	读寄存器 0~15
WRITE	1	1000	A3A2A1A0	写寄存器 0~15
ERASE	1	0100	A3A2A1A0	擦除寄存器 0~15
EWEN	1	1100	0001	写/擦除允许
ENDS	1	1100	0010	写/擦除禁止
* * * WRAL	1	1100	0100	写所有的寄存器
ERAL	1	1100	0101	读所有的寄存器

这里 A3A2A1A0 对应于 16 个 16 位寄存器中的一个。

所有命令、数据输出和输入都是在 SK 时钟的上升沿上移进/移出的。

由 10ms 的 $\overline{CS}$ 低电平脉冲进行写/擦除操作。

所有指令是由 $\overline{CS}$ 上的 LO-HI 转换进行初始化, $\overline{CS}$ 上的 LO-HI 转换之后为 DI 上的 LO-HI 转换。

READ: 在读命令移入后, DI 变为"忽略", 并且可以从空闲位 0 开始从输出上读出。

WRITE: 移入写命令后, 输入数据(16 位), $\overline{CS}$ 的低电平脉冲最小为 10ms。

ERASE/ERASE ALL: 移入命令后, $\overline{CS}$ 保持低电平。

WRITE ALL: 给 $\overline{CS}$ 加 10ms 的低电平脉冲。

ENABLE/DISABLE: 移入命令

在 NMC9306 数据手册中, 对指令系统、E²PROM 时序图及各种考虑进行了详细说明。下面给出 NMC9306 与 COP888CG 接口软件的源程序列表。