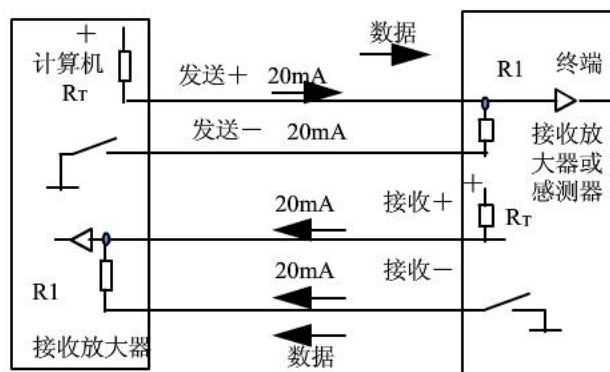


图3 SN75174和SN75175的管脚与内部结构

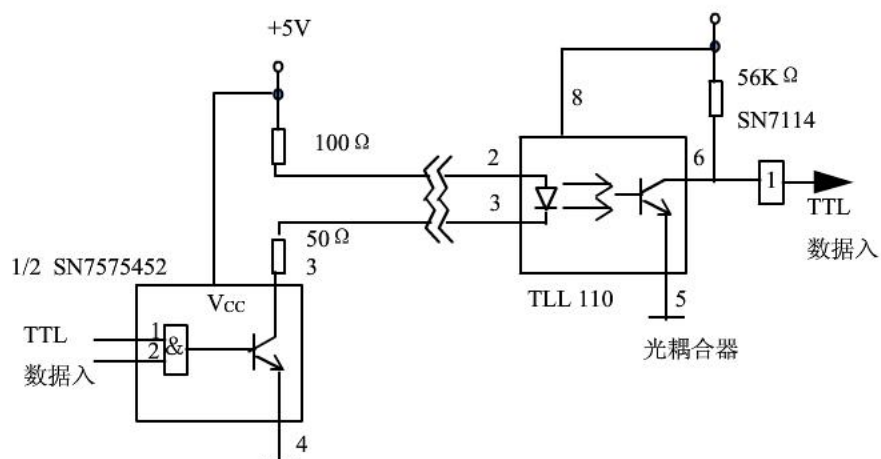
### § 3.3.7 20mA电流环路串行接口与联接方法

20mA电流环路串行接口也是目前串行通信中广泛使用的接口，但未形成正式标准的接口电路。这种接口比RS-232接口简单得多，只有4条线，如图4所示。正、负发送端和正、负接收端形成一个输入电流回路和一个输出电路回路。当发送数据时，根据数据的逻辑1或0，分别使回路形成通断状态，如图4中的开关所示。



20mA电流环路是异步串行接口，每次发送数据时必须以无电流的起始位作为每一个字符的起始位，接收端检测到起始位时便接收一个字符数据。这种接口方法的最大优点是低阻传输线对噪声不敏感，并且容易实现光电隔离。

20mA电流环路串行接口采用信号电流传输，接口电路的联接方法如图 所示。图中先将TTL电路转换成电流信号，传输后经光耦合器再转换成TTL 电平信号。这是一个对地隔离的电流环路传送和接收电路。



本节阐述设计人员在设计TTL电路时需考虑的各种要求。

### 1、电路的极限参数和规范参数

极限参数是保证集成电路能可靠工作的权限值。为了防止电路损坏，电路的工作条件不应超过所规定的极限范围。规范参数是保证集成电路能够正常工作的参数，所以电路必须严格按器件参数规范值进行测试和使用。

目前，国内外推出的各种产品其种类，特性参数，即使是同一种功能的TTL器件其工作特性也不完全相同，因此在选用一种新器件以前，一定要仔细阅读厂家提供的参数。

### 2、电源和地

对电源的纹波和稳定度的要求：

I类、III类  $\leq \pm 5\% (\pm 250\text{mV})$

III类  $\leq \pm 10\% (\pm 500\text{mV})$

同时根据不同系列电路的速度指标和电流变化率的差异还必须加接旁路滤波电容。其容量通常为 $0.01\mu\text{F} \sim 0.1\mu\text{F}$ 。

低频滤波电容器应连接在印制电路板两边电源线入口处，通常用两个 $20\mu\text{F} \sim 50\mu\text{F}$ 的钽电容或电解电容。

### 3、电路的工作环境

国产各系列TTL电路的适用环境温度可分为下述三类：

I类  $-55^\circ\text{C} \sim +125^\circ\text{C}$

II类  $-40^\circ\text{C} \sim +85^\circ\text{C}$

III类  $-10^\circ\text{C} \sim +70^\circ\text{C}$

在此工作温度范围内，电路的功能和参数应完全符合产品规定的要求。

#### 4、电路外引线端的连接

(1) 在使用TTL电路时，不能将电源V<sub>cc</sub>和地线颠倒错接，否则将引起很大的电流而造成电路失效。

(2) 电路的各输入端不能直接与高于+5.5V和低于-0.5V的低内阻电源连接。因为低内阻电源能提供较大电流，会由于过流而烧坏电路。

(3) 不允许将电路的输出端与低内阻电源直接相连接，但可以通过电阻相连以提高输出高电平。当使用集电极开路输出电路时，必须通过计算选用阻值合适的外接电阻。

(4) 具有图腾柱或达林顿输出结构的TTL电路不允许并联使用。只有三态或集电极开路输出结构的电路可以并联使用。当若干个三态逻辑门并联使用时只允许其中一个门处于使能状态（“0”态或“1”态），其它所有门应处于高阻态。当将集电极开路门输出端并联使用时，只允许其中一个门处于低电平输出状态，其它门则应处于高电平输出状态，否则将会出现逻辑错误。

(5) 当将一些集电极开路门路的输出端并联而使电路具有“线与”功能时，通常应在其公共端出端加接一个上拉负载电阻R<sub>L</sub>到V<sub>cc</sub>端。

(6) 集电极开路输出电路中输出管的击穿电压一般在10V以上，有的可达20V。只要在输出管所允许的驱动能力和击穿电压范围内就可任意选用工作电压值，用作电平转换接口，指示灯驱动等。

#### 5、对输入信号边沿的要求

驱动TTL电路的输入信号必须具有较快的转换时间。当输入信号上升或下降时间大于1μs时，有可能在输出端出现信号振荡。这种振荡信号送入触发器或单稳态触发器中就可能引起逻辑错误。

一般组合电路的输入信号上升或下降沿变化速率应小于100ns/V，时序电路输入信号上升或下降沿变化速度应小于50ns/V。施密特触发器把缓慢变化的信号边沿变成陡变的边沿。对于慢边沿输入信号，必须加整形器。

#### 6 TTL 门、扩展器和触发器的使用要求

##### (1)、输出端的连接

TTL电路的输出端不允许与电源短路，但允许瞬间接地。当一个管壳内封装有若干个单元电路时，不允许其中的几个单元电路的输出端同时瞬间接地。

##### (2)、不使用的输入端的处理方法

A • 若电源电压不超过5.5V，与门电路的使用输入端和触发器不使用的置位、复位端可直接连到电源V<sub>cc</sub>上；也可将不用的输入端分别通过一个大于或等于1k $\Omega$  的电阻连到V<sub>cc</sub>上；或把几个不使用输入端通过一个公共电阻连到电源V<sub>cc</sub>上；也可以把同一块电路的不使用的输入端并联到该电路的一个已被使用的输入端上。

B • 如果前级驱动器具有足够的驱动能力，可将与或非门的不使用的与输入端直接连到与或非门的已使用的某一个输入端上，但也可把不使用的或输入端接地。

C • 为了达到最低功耗，可将不使用的与非门和或非门等器件的所有输入端接地，同时也可将它的输出高电平连到不使用的与输入端上。

D • 不使用的输入端可以悬空但决不允许带开路长线，以免产生“低频效应”，造成单拍工作失常。

### （3）扩展器、触发器输入端的连接

带扩展的门应尽量与其所带的扩展器靠近，使其连线最短，以避免增加扩展器连线的分布电容。一个或扩展门最多可带两个或扩展器，而且每带一个或扩展器级延迟将增加40%左右。

不应将触发器的不使用输入端悬空。触发器的输入端连接线应尽量短。当用若干触发器构成时序电路时，由于时钟信号沿传输线传送时存在着时间偏差，到达或进入触发器的时钟信号将产生时钟偏移 $t_{skew}$ ，应使 $t_{skew}$ 尽可能小。

### （4）、触发器对输出负载和信号源的要求

A 应正确区分锁存触发器，D型触发器和J-K触发器对电位输入和时钟信号的要求，即电位输入信号（数据）与时钟信号的时间关系必须满足推荐使用条件中的建立时间 $t_{set}$ 和保持时间 $t_h$ 的要求。

B 一般不允许触发器的输出直接驱动各种指示灯、长传输线和电感负载，否则必须附加缓冲门。

## 三 CMOS电路的工程设计

CMOS IC是压敏器件，其输入阻抗很高，用CMOS IC直接带动CMOS IC时，其直流负载是很轻的。主要考虑电平匹配就可以。

### 1、对总线终端的驱动

高速CMOS系统中，以CMOS IC驱动总线是很常见的事。工作时不能让总线浮空，而是通过上拉或下拉把总线接到V<sub>cc</sub>或V<sub>ss</sub>上。为了提高工作速度，根据一般充电慢于放电的特点，在实际中大多数采用常态时将总线上拉为高电平的方法。加入上拉或下拉电阻后，可以大大减少总线受噪声的干扰，使总线工作在所有三态总线驱动器全部处在三态时，也不会被悬浮起来的状态。



DSP有自己的汇编语言指令系统，为适用于高级语言编程，各公司也陆续推出适用于DSP的高级语言编译器，主要是C语言编译器，也有Ada、Pascal等编译器，程序员可用高级语言编程，通过编译器，将程序汇编、连接成DSP目标代码。

## 2、DSP应用领域

随着DSP性能的改善和成本的降低，DSP在通用数字信号算法处理的基础上，其应用几乎遍及电子学每个领域。

(1) 通信：网络通信，高速调制解调器，编/解码器，自适应均衡器，移动通讯，语音邮箱，回音消除，噪声对消，会议电视，扩频通信等。

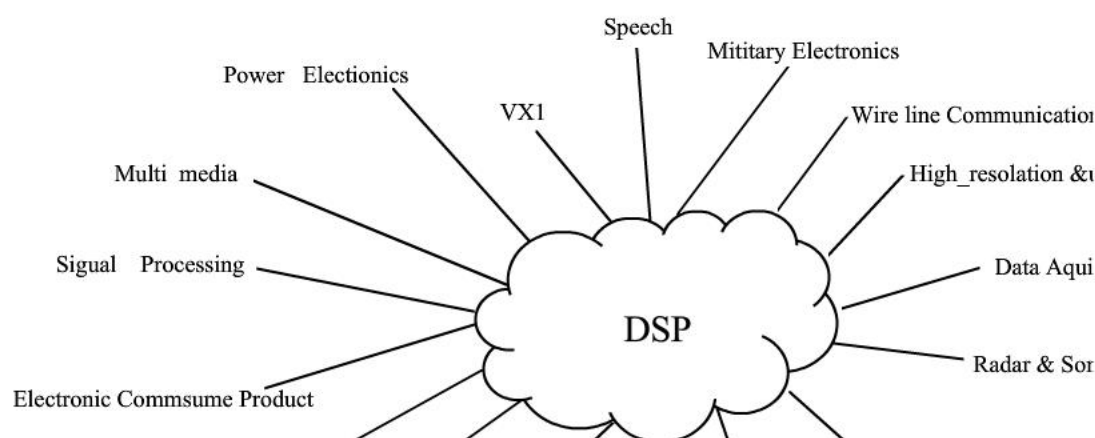
(2) 声音/语音处理、图像/图形处理：语言识别，语言合成、语音编码、图形变换、机器人视觉、模式识别、电子地图等。

(3) 仪器/仪表：谱分析、函数发生器、波形发生器、数据采集、状态分析、石油/地质勘探、VR航空（Visual Reality）试验，医学仪器等。

(4) 计算机：阵列处理器、图形加速器、神经网络、多媒体等。

(5) 消费电子：数字电视、高清晰度电视、VR游戏、汽车工业等。

(6) 军事：雷达与声纳信号处理、导航、制导、全球定位、搜索跟踪、VR式器试验等。如下图：



(下面以TMS320C54X系列为例介绍DSP的硬件结构及软件编程方法)

### § 3.10.3 TMS320 C54X DSP硬件结构

#### 1、TMS 320C54X内部组成单元与总线结构

TMS320C54XDSP的结构简图如图1所示：

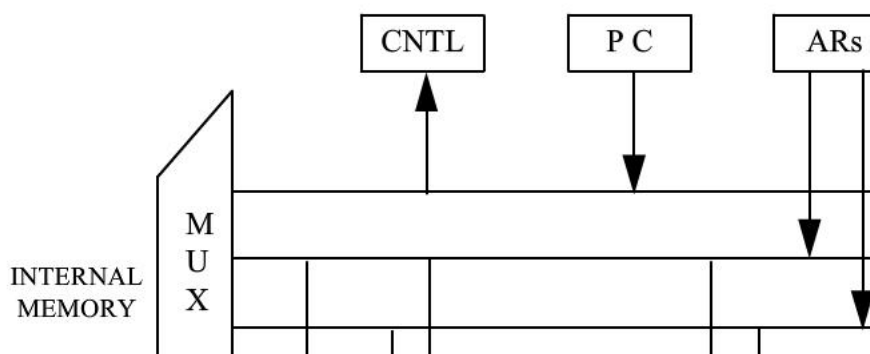


图1 TMS320C54X内部组件与总线连接

可见DSP内部的程序总线P、数据总线D、控制总线C和数据写存储器总线E都是独立分开的，以便于组件单元的并行处理和数据传送。

TMS320C54X C54X的结构部件其实可看作是围绕挂换在8束16 bit的总线上的。（见图2 TMS320 C54X内部硬件结构框图），其中：

- （1）程序总线PB传送指令码和程序存储器的直接操作码。
- （2）三束数据总线（CB、DB和EB）相互连接起各个单元，如CPU、数据地址产生逻辑（DAGEN）、程序地址产生逻辑（PAGEN）及其它芯片组件和数据存储器（其中CB、DB传送从数据存储器读出的数据，EB传送向存储器写的数据）。
- （3）四束地址总线（PAB、CAB、DAB和EAB）传送指令执行期间所需的地址数据。

## 2、内部存储器结构

54X的存储器由程序空间、数据空间和输入/输出空间三个分开的空间组成，包括随机存取存储器RAM和只读存储器ROM。RAM可当作DARAM（dual-access RAM）和SARAM（Single-access RAM）54X系列中各种芯片的ROM、DARAM和SARAM空间大小有所不同。

- （1）片上ROM（On-Chip ROM）它是程序存储器的一部分，常用以接受boot Loading。
- （2）DARAM（On-Chip Dual-Access RAM）。它由若干块组成，对DARAM中的数据，CPU在一个机器周期内可对它进行两次操作（读或写），DARAM总是映射在数据空间，其主要目的是存储数值，它也可映射于程序存储空间用以存储程序代码。
- （3）SARAM（On-Chip Single-Access RAM）。它由若干块组成，每块在一个机器周期内，可被读或写一次，它一般映射在数据空间，主要是存储数值，它也可映射于程序空间存储程序代码。
- （4）片上RAM保护。当设计片上RAM保护时，外部指令就不能对其进行读写。

## 3、CPU（Central Processing Vnit）

TMS 320 C54X CPU包括：

a) 40-bit的算术逻辑单元 (ALU)。可实现2个40bit操作数的算术逻辑运算, 也可进行布尔运算。

b) 2个40-bit的累加器 (A、B)。可存储ALU或乘/加 (Multiplier/adder) 运算的输出, 也可提供ALU的二次输入。累加器A还可作乘/加运算的输入, 每个累加器都可划分成以下三部分:

i) 保留位 (bit 39-32);

ii) 高字节 (bit31-16);

iii) 低字节 (bit15-0)。

c) 桶式移位器 (Barrel shifter)。桶式移位器有一个来自40bit的累加器 (A或B) 或数据存储器 (通过CB或DB) 的输出, 其40bit的输入, 送到ALU或到数据存储器 (通过EB) 中。它可对输入产生0—31bit的左移和0~16bit的右移。并由指令进行控制。

d) 17×17-bit的乘法器。它实现两个17bit操作数的乘运算, 并与40bit的加法器等共同实现在一个指令周期内完成乘/加运算; 和AKU共同在一个指令周期内完成并行的乘/累加 (MAC) 运算。

e) 40-bit加法器 (Adder(40)), 配合Multiplier执行乘/加运算。

f) 比较、选择和存储单元 (CSSU), 实现累加器高、低字节中数据的大小比较测试或控制标志位TC、状态寄存器ST0和发送寄存器 (TRN) 以保持发送状态等。

g) 数据地址产生单元 (DAGEN)

h) 程序地址产生单元 (PAGEN)

#### 4、DSP外部信号引脚

DSP处理信号引脚包括: (C548)

(1) 外部总线接口包括: (External Bus Interface)

并行地址总线口 (23-bit)

并行数据总线口 (16-bit)

存储器选择MSTRB\_、程序空间选择PS\_、数据空间选择DS、I/O选择IOSTRB\_、IS\_读/写信号R/W、READY、信号保持HOLD\_、指令获得IAQ\_等。

(2) 控制信号 (Control Signals)

复位信号RS;

外部中断输入INTX\_及中断响应IACK\_等;

分支控制BIO-;

外部标志输出XF;

工作方式选择MP/MC等。

(3) 晶振/时钟信号 (Oscillator/Timer Signals)

晶振输出插入 X1、X2/CLKIN



时钟信号：包括CLKOUT、CKKMDX、TOUT等。

(4) 缓冲串行口信号 (Buffered Serial Ports Signals)

接收时钟：BCLKRO、BCLKR1；发送时钟：BCLKXO、BCLKX1；

串行数据接收：BDR0、BDR1；串行数据发送BDX0、DBX1；

帧同步接收：BFSR0、BFSR1；帧同步发送BFSX0、BFSX1。

(5) TDM串口信号

TDM 接收时钟TCLKR；TDM发送时钟TCLKX

串行数据接收TDR；串行数据发送TDX

接收帧同步/地址 TFSR/TADD；帧同步发送TFSX/TFRM

(6) 仿真信号 (Emulation)

JTAG测试时钟TCK

JTAG数据输入TDI

JTAG口模式选择TMS

JTAG数据输出TDO

JTAG口复位TRST\_

仿真引脚 EMU0、EMU1/OFF

(7) Host接口Host Port Interface

并行数据口 HD0~7

控制输入 HCNTL0、HCNTL1

字节识别输入 H+BIL

片选 1+CS

数据输入选择 HDS1\_HDS1；

地址选通：HAS；

读/写信号：HRW、HRDY

串断输出：HINT；

HPI模式选择 HPIENA/VDD

如图3 “C54X信号引脚”和“C548信号引脚”。

## 5、DSP硬件设计的几个主要问题：

(1) 总线控制方案

(2) boot loading

(3) Hardware wait\_states

(4) I/O与中断设计

(5) Memory Map

## (6) 串口工作方式与时序设计

## (7) TAP接口

### § 3.10.4 TMS320C54X的软件编程

#### 1、DSP编程工具与流程

DSP的设计目标是进行数字信号处理，在硬件设计的基础上选择好一定的优化算法并通过编程在DSP芯片上实现是DSP技术的核心内容。对DSP进行编程，目前最有效的语言工具仍是DSP汇编语言，同时为方便用户用高级语言进行编程开发，也相继推出了C语言编译器，ADa、Pascal等编译器。

图4是TMS C320C54X软件开发流程，图中有阴影的部分是通常开发的必需步骤，其它可根据需要选作，最后产生COFF（Common Object File Format）文件。

#### 2、DSP汇编语言编程

##### (1) DSP汇编过程有以下几个步骤

a) 源程序的编写。将算法的详细实现过程用DSP指令系统的语句描述出来，其中源程序一般包括：data、bss和text三个基本组成内容。下面是一段程序例子，图5是COFF文件格式和object file在存储器中的分配情况。

```
2          ****
3          ** assemble an initialized table into data.**
4          ****
5 0000                      data
6 0000 0011      coeff word 011h,022h,033h
7          0001 0022
8          0002 0033
9          ****
10         ** Reserve space in bss for a variable **
11         ****
12         bss  buffer,10
13         ****
14         ** Still in .data.
15         ****
16 0003 0123      ptr word 0123h
17         ****
18         ** Assemble code into the text section ****
19         ****
20 0000          text
21 0000 100f      add: LD  0Fh,A
22 0001 f010      aloop: SUB  #1,A
23         0002 0001
24 0003 f842      BC  aloop,AGEQ
25         ****
26         ** Another initialized table into data ****
27         ****
28         data
29 0004          ival: .word 0AAh,0BBh,0CC'h
30         0005 00bb
31         0006 00cc
32         ****
33         *** Define another section for more variables**
34         ****
35 0000          var2 .userct "newvars",1
36 0001          inbuf .userct "newvars",7
37         ****
38         *** Assemble more code into text ****
39         ****
40 0005          text
```

Routing Effort ( )1--4

B.Use Xact-Performance ( )

## 2.Optimization

( )Trim Unconnected Signals

( )use Global Resources For High Fan-out Signals

( )Create RPMs for Register-based X-Blox Modules

## 3.Guide/Resource

A.Resources Available for Routing

CLBs:( )partially used ( )Any

Unused global Buffers

B.Guide Placement

( )All Blocks ( )Only block that Have Routed Signals

C. Guide Routing

Locking Routing ( )None ( )Whole Signals ( )All

## Xilinx FPGA XC5200 Configuration

### 1.Configuration

Inputs threshold:( )TTL ( )CMOS

configuration Rate:( )Fast ( )Slow ( )Wed

configuration Pins:

Program:( )float ( )Pull-up

Done: ( )Float ( )Pull-up

Perform CRC During Configuration ( )

Produce ASCII configuration File( )

### 2.Startup

Startup clock :( )clock ( )user clock

( )Synchronize Startup to Done IN Pin

Output Events:

Done:( )C1 ( )C2 ( )C3 ( )C4

enable Outputs :( )C2 ( )C3 ( )C4

Release Set/Reset: ( )C2 ( )C3 ( )C4

### 3.Readback

Clock :( )clock ( )user clock

( )Capture CLB/IOB Outputs when TRG Goes Active

( )Abort Readback when TRIG Goes Inactive

Optional Targets

☐ Produce Timing Simulation Data

☐ Produce Timing Report

☐ Produce Configuration Data

## 2.Startup

Startup clock : ☐ clock ☐ user clock

☐ Synchronize Statuip to Done IN Pin

Output Events

Done: ☐ C1 ☐ C2 ☐ C3 ☐ C4

Enable Optputs: ☐ C2 ☐ C3 ☐ C4

Release Set/Reset : ☐ C2 ☐ C3 ☐ C4

## 3.Readback

Clock ☐ clock ☐ user clock

☐ Capture CLB/IOB Outputs when TRG Goes Active

☐ Abort Readback when TRIG Geses Inactive

## Optional Targets

☐ Produce Timing Simulaton Data

☐ Produce Timing Report

☐ Produce configuration Data

## 2.Startup

Startup clock : ☐ clock ☐ user clock

☐ Synchronize Startup to Done IN Pin

## Makebit

— I (Ignore Critical Net Flags on Nets)

— U (Enable Mackebits to Tle using critical nets)

— B (Create rawbits file in RBT)

— C (Cmos inputs)

— D (Create DRC report in DRC)

— J (Suppress making bitstream)

— L (Create Readback into file in .LL)

— M (Creare mack file in .MSK)

— MBO=(File) (Read Makebits options from FILE)

— N (Save tied desigt)

— O (Output filename)

— P (No pullup on LCA Done pad)

- R0 (Disable readblack)
- R1 (Enable readback once)
- R2 (Enable readback many)
- S0 (Disable Oscillator)
- S1 (Enable Oscillator)
- S2 (Enable Oscillator Div 2)
- T (Tiedown Unused Interconnect)
- V (verbose Mode)
- XA (Done after IOBs active)
- XB (Done before IOBs active)
- YA (Reset after IOBs active)
- YB (Reset before IOBs active)

#### (B) ViewLog DOS版本

Xmake的部分选项

- X Use XNF files only
- L Use old library only
- B perform XBLOX optimization
- I 〈file〉 Use PPR Guide file
- G Generate MAK file only
- O Don't redirect output
- R Force re-execution of all programs
- V Verbose message made

#### Makebit

- Input TTL ( ) COMS ( )
- Donepad Pullup ( ) NoPullup ( )
- Read 0 ( ) 1 ( ) Cmd ( )
- XtaIOsc Disable ( ) Enable ( ) DIV2 ( )
- DoneTime Before ( ) After ( )
- RestTime Before ( ) After ( )