

语言芯片植入继电保护的方法

顾冰

顾思远

姜勇

阿城继电器二开分厂(150302)

阿继集体公司(150302)

阿继研究所(150302)

摘要 本文首先介绍了目前语言电路的发展、构成、原理。阐述几种适用于继保领域的语音芯片和存储芯片的性能特点,最后介绍语音编程器的原理、使用及语音电路植入继电保护装置中的方法。

关键词 语音描述 智能质数 语音处理 语言植入

目前继电器和继电保护装置的动作指示和故障报警,多靠的是LED、灯泡、光字牌、蜂鸣器电笛、喇叭的声光指示。随着技术的发展,出现了依靠CRT来显示故障地点、时间、继电器动作情况的自动装置,能不能用语言来描述故障所在和装置动作情况?特别是故障时提醒运行人员操作处理或更进一步以语言形式作出趋势预报?回答是肯定的。随着固态语音芯片和超大容量存储器的出现,将使继电器智能化达到新的水平。为此我们将适合于继电保护的语音电路与合成方法展示于下,供广大同仁参考。

1 语音合成方法与原理

语音芯片已广泛应用于行车、银行,究其原理实际上是一个固态录音机,这种录音机价格低,加之没有机械转动部分,不易损坏,适合用于继保领域取代蜂鸣器、电喇叭等。语音芯片系统包括输入语音处理、存储器、输出语音处理几部分,其关系框图如图1所示:

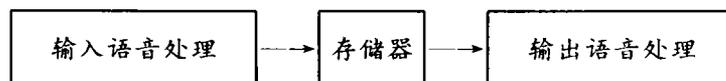


图1

语音处理的方式有很多,常见的有以下几种:

(1)波型合成法(PCM式)就是把语音波形的各点用模数转换器(A/D)变成数字量,存到存储器中,需要时再通过数模转换器(D/A)复原成语音波形经放大至扬声器放出原来的语音。

(2)自适应增量调制法(称ADM式)实际上是PCM式的改良法,但因其数码率较低20kb/s,所以适合于较长的语音录制用。

(3)参数合成法:它是把语音的表征提出来存储入存储器,即用语音的表征来表现语音,这种语音合成法的数码率更低,其值为1.2kb/s,可以用很少的存储器容量存大量的语音。

几种语音处理方式的优劣比较见表1

收稿日期:1994-08-05

表 1

处理方式	数码率	音质	语言长短	参考价格	方便程度	应用情况	注
PCM	64kb/s	清晰	短	低	容易	较广	
ADM	20kb/s	较清晰	较长	较低	容易	较广	
参数合成	1.2kb/s	较清晰	长	较高	较容易	较少	

2 语音存储器

语音存储器分为只读存储器和读写存储器。只读存储器常用 EPROM 或写入方便价格低廉的 Flash 存储器。读写存储器则采用静态的 SRAM, 动态的 DRAM 等。

因读写存储器怕掉电, 不适合用于要求长期保留的语音场所。只读存储器中的 ROM 型存储器因写入困难, 所以大都采用 EPROM 型存储器。考虑到存放语音存储量较大, 我们常用的 2732、2764 等仅有几 K 容量的存储芯片就不够用了, 尤其要说几句不同内容的话常作分段使用, 所以常需十几秒或几十秒的语音段, 为此我们展示常用的只读存储器的性能于表 2, 供参考。

表 2

EPROM 型号	容量	总时间 (s)	最多可分段数 (段)	每段容量 (k)	每段时间 (s)	参考价格 (元)	注
27040	4M 位	256	128 段	32	2	约 65	
27020	2M 位	128	64 段	32	2	约 50	
27010	1M 位	64	32 段	32	2	约 40	
27512	64k 位	32	16 段	32	2	约 25	
27256	32k 位	16	16 段	16	1	约 18	

此外 271001、272001 等可擦除的只读存储器, 也有应用, 其技术参数请参考有关书籍。

3 语音处理器

为了减少芯片数量, 录放(即语音处理)多采用一个芯片, 加少量的外围元件组成。这样的芯片如 YYH402、YYH403、T6668、SR8803 等, 有关参数见表 3。

表 3

型号	语音处理方式	存储器种类	常用存储量	录放时间 (s)	注
YYH402	ADM	SRAM EPROM	512k 位	32	
YYH403	ADM	SRAM EPROM	512k 位	32	
T6668	ADM	DRAM	1M 位	60	
SR8803	PCM	ROM			制专用大量芯片

4 语音电路的组成

以适合于继电保护应用的语音电路为例说明如下: 它是由语音处理芯片 YYH402 和存储器 27256, 加少量的外围器件和放音电路组成的如图 2 所示, 放音时间可长达 40s。

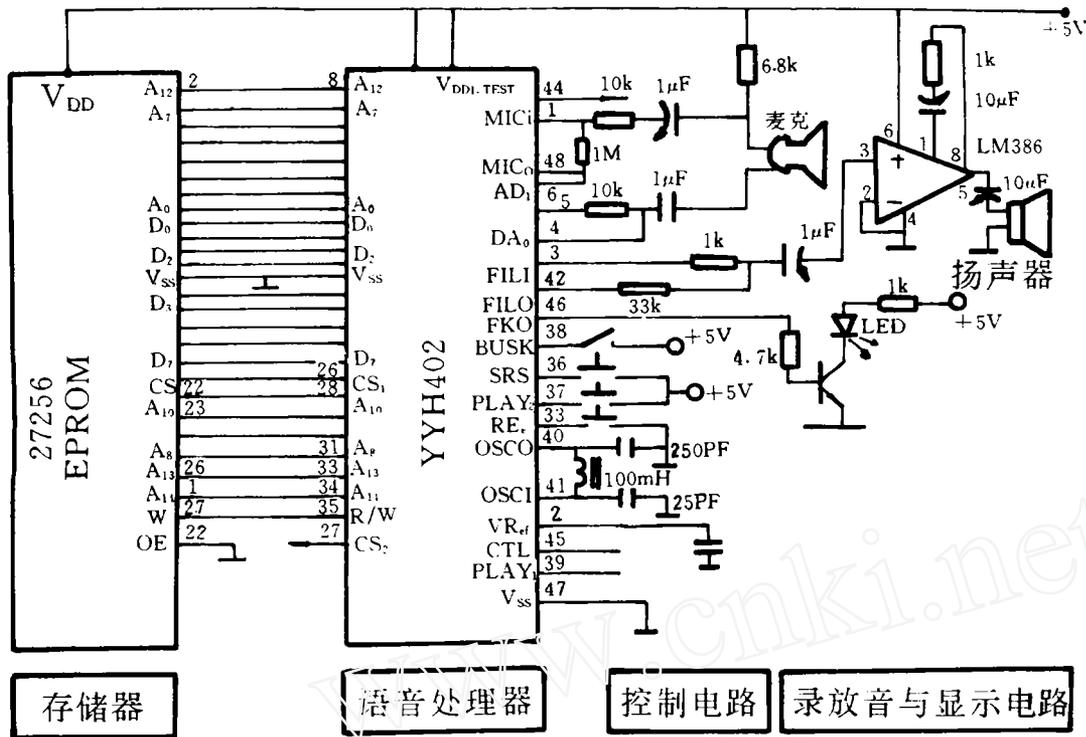


图 2

图 2 中如把 27256 换成 62256 随时可以录音,只要按动 REC 键 LED 亮把音声送入麦克,经 YYH402 进行语音处理后存入 62256。当需要放音时,按动 PLAY₂ 键,LED 亮时存在 62256 中的语音信息便从 FILO 送到 LM386 中,经功率放大后驱动扬声器。如果需要重复放音按动 PLAY₂ 即可,可重复任意次。不需要时可断电清除。需要永久保持时,可采用不怕掉电 27256 大容量 EPROM 系列芯片。但烧录就麻烦了一点,需要专门的语音烧录的编程器。

实际我们需要的是能说几段话的语音电路如某机组保护提醒运行人员:《主保护差动已跳闸》《异常运行的失磁保护已跳闸》或《××已有××异常运行趋势》导语句,这样就需要一个语音电路分四段或者多段。利用图 2 的电路可把语音处理芯片换 YYH403,存储器的容量可以扩大如 27010 等即可实现分四段的愿望。但因 YYH403 没有暂停信号,所以限制了它的应用范围。我们常借助于分段器 FD256 来分段,就可以任意分段了,把图 2 略加改动加上分段器 FD-256 即可分成四段如图 3 所示,图中发光二极管通过存储器 27256 的信号线 A₁₃、A₁₄,给出 00、01、10、11 四段来显示录放音的段数。其它电路和图 2 大致相同。如果需要分的段数再多,存储器就需增大容量,如分成 32 段,存储器改成 27040 是 4M 位的 EPROM 芯片,如图 4 所示,这样看,一个二次系统所要说的话基本可以够用了(即有 32 段话)但是还不能尽我们的心意,就是还不能按我们要求的任意时间,放出其中的一段,它只是能在 YYH402 的 STOP 控制下按顺序放出 32 段,每段如 64k 位 STOP 停一次,这样只能达到段落清楚,要想任意的放出某一段,可以靠时间严格分离或者以靠其它电路或计算机来控制分段器 FD256 的 T₁~T₅ 的电位而得到任意段的发音。

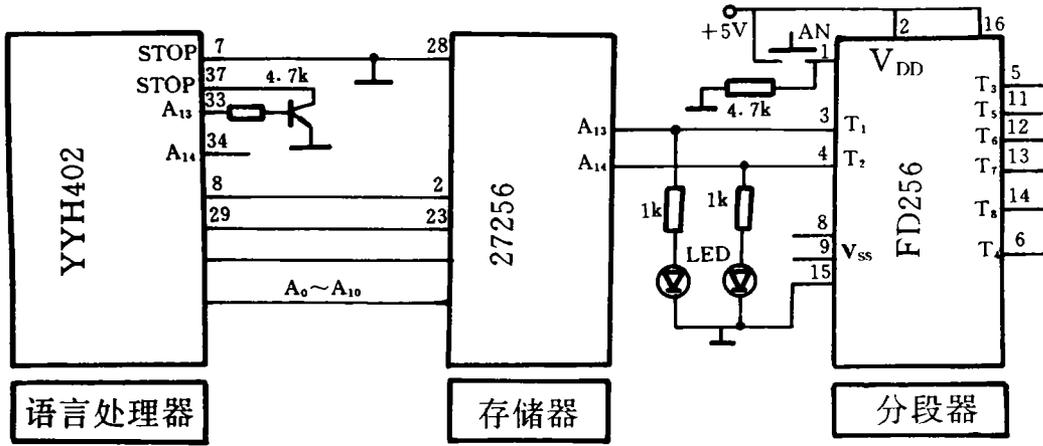


图 3

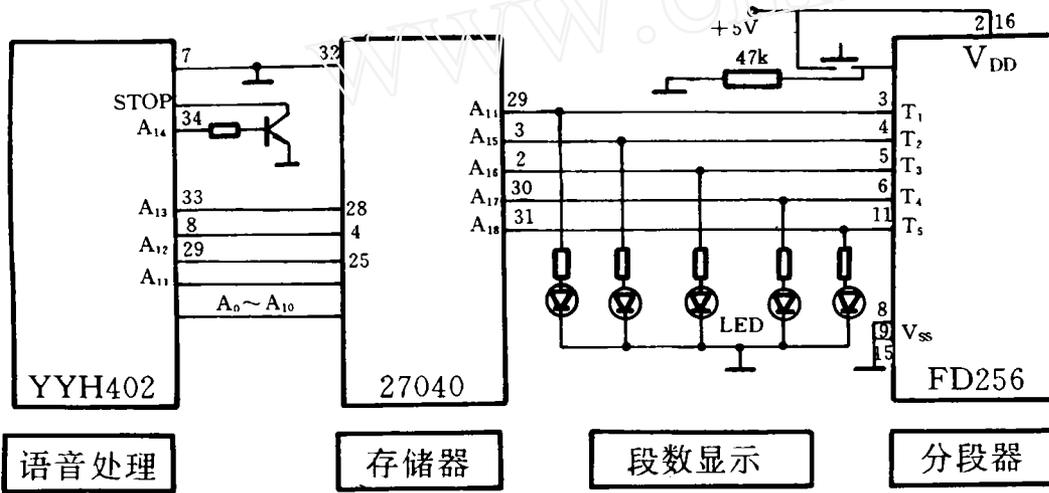


图 4

5 语音烧录法

前面提到了不怕掉电的 EPROM 存储器,语音是永远保存。但 EPROM 必须在 12.5V 的电压下才能录入,而且还须复杂的时序和信号控制,常借助于计算机(或单片机)的控制来烧录,其框图如图 5 所示,通过单片机 8098 执行存在 2764 中的烧录程序,把麦克输入的语音信号处理后暂存在 62256 中,再在 CPU 的控制下烧录到 27256 中,把烧录好的 27256,经放音试验无误,即可从烧录板上取下,插在图 2 中 27256 的位置上就可以执行我们需要的任务了。

我们还可采用成品语音烧录器如 SR92000 系列。特别是 SR92006 算是档次较高的语音编

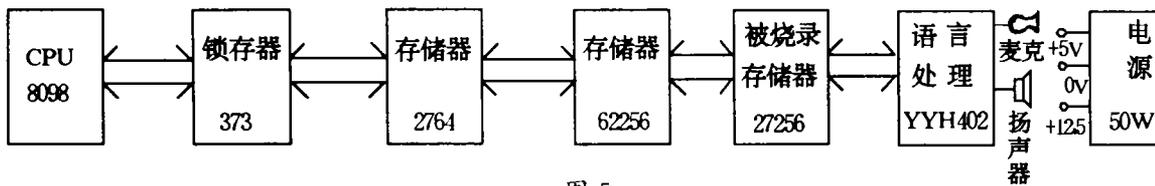


图 5

程器了,其操作简便,功能较强。

6 语音电路植入继电保护

语音电路(如图 2 的电路)植入继保装置中的方法比较简单,直接接入继电保护中,用出口继电器的触点控制放音按键即可,当继电器动作后,触点闭合,开始放音(如提醒运行人员某继电器已经跳闸)。

但是把语音电路植入哪些继电器是值得讨论的,若所有继电器及自动装置都说话,势必要显得混乱、嘈杂,这时我们希望重要保护会说话,报告故障所在,或是故障趋势,以便快速处理,把事故减到最小程度。现在大系统大多有管理(监控)计算机,多种保护继电器都在计算机监控下工作,建议把语音电路(特别是多段语音电路)也置于计算机的监控下,什么时候放音什么时候放哪段语音,指示哪种故障,由计算机统一管理。

结束语:在国外语音电路已很普及了,各种仪器、仪表、设备都有语言报数、越限报警、故障报告等。在这方面我国技术也逐渐成熟了,希望能在继电保护中尽快采用。提高继电器及自动装置的智能质数。现在新出现 Flash 存储器能象 E²PROM 一样可在 +5V 电压下烧录,是靠电擦除的不怕掉电的存储器、价格比 E²PROM 低很多,容量也很大,如一片 20F001 就有 1M 位的存储空间,可满足各种电力系统的需要。

可 编 程 大 功 率 电 子 负 载 器

2 月 20 日在 811 研究所召开可编程大功率电子负载器等新产品鉴定会。会议出席有航空工业总公司,611 研究所,811 研究所,809 研究所,上海现代电源设备公司,上海工业自动化仪表研究所和国家有关部委 20 多人。代表们审查新产品各项技术特性和文件,并观看操作表演。会议认为编程大功率电子负载器硬件可靠,软件使用方便,显示清晰,达到了先进水平。会议还审查了可编程大功率稳压电源,可编程大功率精密电阻和大功率电源等。

可编程大功率电子负载器允许通过电流 0 至 1000A,最大耗散功率 20kW,负载器对电源放电方式由计算机控制,恒流方式或脉冲方式或按设计曲线放电。计算机精确地自动记录储存放电过程中电源电压,电流及电池温度的变化。由打印机输出各项数据和曲线。可编程负载的各类放电方式很容易评估各种电池的内在质量,电池的各种工艺变化带来的影响,也可以对各类电源的负载特性和响应时间等作出精确显示。它又是对各种电气接触研究的有效设备。

现代电源公司 姚善宝