

## ◎产品、研发、测试◎

## 一种高性能 DSP 专用地址产生器的设计

单超,沈绪榜,王卫涛

SHAN Chao, SHEN Xu-bang, WANG Wei-tao

西北工业大学 计算机学院, 西安 710072

College of Computer, Northwestern Polytechnical University, Xi'an 710072, China

E-mail: shanchao55@163.com

SHAN Chao, SHEN Xu-bang, WANG Wei-tao. Design of high-performance special address generator for DSP. Computer Engineering and Applications, 2007, 43(33): 98-101.

**Abstract:** DSP(Digital Signal Processor) is fit for the field of digital signal processing with the requirement of high performance of real time. This paper proposes an AGU (Address Generation Unit) which can satisfy the special addressing modes of DSP. This AGU also can supports parallel instruction, and realize the operations of arithmetic and address in parallel, which improves the digital signal processing speed efficiently.

**Key words:** address generation unit; addressing mode; digital signal processor

**摘要:** DSP 处理器面向数字信号处理领域, 具有高度的实时性要求。论文设计了一种能够满足 DSP 处理器特殊寻址方式的地址产生单元, 同时支持并行指令的执行, 实现了算术运算与地址运算的并行处理, 有效地提高了数字信号的处理速度。

**关键词:** 地址产生器; 寻址方式; 数字信号处理器

**文章编号:** 1002-8331(2007)33-0098-04 **文献标识码:** A **中图分类号:** TP302

## 1 前言

与单片机相比, DSP 芯片具有更加适合于数字信号处理的软件和硬件资源, 可用于复杂的数字信号处理算法。尤其在有一些特殊的场合, 要求的信号处理速度极高, 例如专用于 FFT、数字滤波、卷积、相关等算法的 DSP 芯片, 需要频繁地改变数据地址, 而传统处理器需要几个指令周期才能改变数据地址, 无法满足数据处理的实时性要求。因此几乎所有的 DSP 芯片都有和地址寄存器联系的算术或逻辑功能块, 即地址运算单元, 地址运算单元和算术运算单元并行执行以改变地址。

然而, 传统的地址产生器只能完成地址简单的加、减操作, 其计算地址方式单一, 不能够满足 DSP 芯片复杂的数据处理要求。根据常用的数字信号处理算法, 常用的寻址方式除顺序寻址方式外, 还包括位翻转寻址以及循环寻址。文章设计的地址产生器很好的支持了上述三种寻址方式, 除此之外, 该地址产生器支持双地址线的并行计算, 从而很好的支持了并行指令的执行, 有效的提高了数据处理的并行性, 大幅度提高了 DSP 处理器性能。

文章首先介绍了 DSP 常用及专用的寻址方式, 之后给出了该地址产生器各种寻址方式的算法, 以及对该地址产生器进行了逻辑实现。文章最后对该地址产生器的体系结构进行了介绍, 并且进行了性能分析。

## 2 DSP 处理器中地址产生器的寻址类别

## 2.1 DSP 处理器常见的寻址方式

DSP 处理器要对大量的数据进行实时性处理, 一般都有强大的寻址方式与之对应, 能够从存储器、寄存器和指令编码中取得数据。常见的寻址方式如表 1 所示。

表 1 DSP 处理器常见的寻址方式

寄存器寻址	操作数存放在寄存器中
直接寻址	数据地址由数据页(DP)寄存器内容与部分指令字编码组成
间接寻址	操作数地址由寄存器存放或者由寄存器与偏移量相加产生
立即数寻址	操作数即为指令字的一部分, 无需寻址
.PC-相关寻址	指令字地址由当前 PC 值与偏移量相加产生

立即数寻址无需进行操作数寻址, 操作数包含在指令字中。寄存器寻址和直接寻址无需进行地址运算, 利用指令字和简单的指令译码控制就能够完成寻址。PC-相关寻址虽然要进行地址运算, 但是只有在遇到指令跳转时才会用到。用到的间接寻址是 CPU 最常用的寻址方式, 并且在寻址时需要对地址进行算术运算处理, 如下所示:

$$Address = (reg) \pm (IR)$$

$$Address = (reg) \pm immediate\_value$$

其中, reg 寄存器中存放基地址, IR(Index Register)寄存器中存放变址, 操作数的目的地址(Address)为基地址相对于变址或

**基金项目:** 国家部委预研基金项目。

**作者简介:** 单超(1982-), 男, 硕士研究生, 研究方向: 计算机系统结构、专用集成电路设计; 沈绪榜(1933-), 男, 研究员, 博士生导师, 中科院院士, 主要研究方向: 计算机体系结构设计、嵌入式计算机及其芯片的国产化实现设计、VLS芯片与智能计算机设计; 王卫涛(1981-), 男, 硕士研究生, 研究方向: 计算机系统结构、专用集成电路设计。

者立即数的偏移地址。

### 2.2 DSP 运算专用的寻址模式

为了更好地支持数字信号处理算法如 FFT、卷积运算等,通常 DSP 处理器还支持位翻转寻址方式和循环类寻址方式。

DSP 支持位反转寻址方式,主要是为了完成快速傅立叶变换(FFT)。FFT 变换过程:数据表中顺序存放的数据经过若干级蝶形运算后对应结果以位翻转顺序存放在数据表中。

从图 1 可以看出,输入为混序排列,运算为 3 级,每级 4 个蝶形运算,在 PC 机上实现时,把  $x(0)\cdots x(7)$  相应分配  $M(0)\cdots M(7)$  8 个单元,每级运算完后各存储单元原来的值不再使用,那么蝶形运算结果仍可保存在  $M(0)\cdots M(7)$  中,依次类推,最后结果为顺序排列。可以看出,输入序号二进制码是输出序号二进制码的按位翻转,要使输入输出相对应,就需用子程序实现,但是若设置位翻转寻址方式就不需要花费很多额外的指令周期,大大加快了 FFT 实现速度。

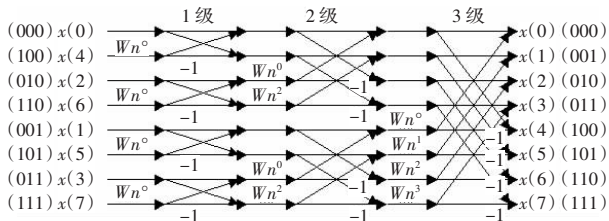


图 1 8 点时间抽选的 FFT 流图

位反转寻址的规则为进位位向低位进位,例如初始时  $AR_n$  为 0110\_0000,  $IR$  为 1000 时,寻址过程如表 2 所示。

表 2 位反转寻址模式

位反转寻址指令	位反转寻址后 $AR_n$ 结果
$AR_n ++ B(IR)$	0110_1000
$AR_n ++ B(IR)$	0110_0100
$AR_n ++ B(IR)$	0110_1100
$AR_n ++ B(IR)$	0110_0010
$AR_n ++ B(IR)$	0110_1010
$AR_n ++ B(IR)$	0110_0110
$AR_n ++ B(IR)$	0110_1110

对于循环类寻址方式,相关运算或者卷积运算需要在存储器中设置一个长度为  $L$  的循环缓冲器。循环缓冲器作为一个滑动窗口,保存最近要处理的数据。当新数据到来时,数据指针将对循环缓冲器定位,如果数据指针到达循环缓冲器尾部时,数据指针将重新指向循环缓冲器首部,新数据将替代老数据。如图 2 所示:设循环缓冲器长度为 6,步长为 1,总共读进 8 个数据,开始显示了读进 4 个数据后的循环缓冲器的情况,之后显示了读进 8 个数据后循环缓冲器的情况。

### 3 地址产生器的生成算法

根据第 2 章总结的数字信号处理用到的间接寻址、位反转

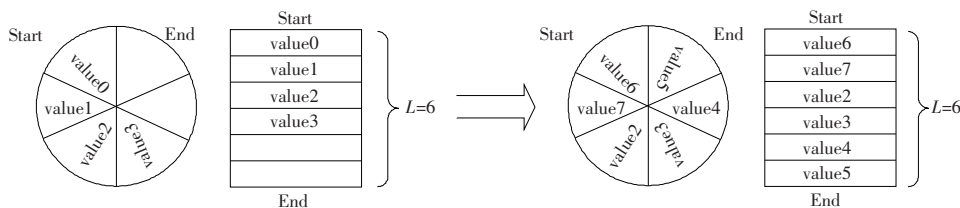


图 2 循环寻址数据存储的逻辑及物理示意

寻址、循环寻址的计算方法,地址产生器的生成算法描述如下所示:

```

Int Agu_algorithm(mod, AR ,disp, IR, step, BK, START, END)
{int mod; /* mod: 寻址类别标识 */
 int AR, disp, IR, step, BK, START, END;
 switch(mod)
 {case 1: AR=AR;break;
 case 2: AR=AR±1;break;
 case 3: AR=AR±disp;break;
 case 4: AR=AR±IR;break;
 case 5: AR=B(AR+IR);break;
 case 6: {AR=AR+step;
 if(AR>END)
 AR=AR-BK;
 else AR=AR}
 case7: {AR=AR+step;
 if(AR<START)
 AR=AR+BK;
 else AR=AR}
 default: AR=AR;
 }
 }

```

其中  $AR$  为 DSP 中的辅助寄存器,存放基地址,  $IR$  为索引寄存器,存放变址,  $disp$  为立即数,  $step$  为循环寻址的步长,  $START$  和  $END$  为循环寻址的起始和结束地址,  $BK$  为循环缓冲区大小。在地址产生器生成算法中,  $case1$ 、 $case2$ 、 $case3$ 、 $case4$  表示顺序地址计算,即间接寻址。  $case5$  表示位反转寻址,其中  $B(AR+IR)$  表示反位相加。  $case6$  表示循环递增地址计算,  $case7$  表示循环递减地址计算。

### 4 地址产生器的逻辑实现

该地址产生器总共有三条数据路径,分别对应三类寻址方式计算,如图 3 所示。其中,变址选择逻辑用于选择变址形式,对立即数和索引寄存器进行选择判断。

地址运算逻辑是地址产生器的主体逻辑,它能完成顺序寻址地址计算与倒位序寻址地址计算。其中,顺向加法与位反向加法如下所示:

顺向加法:  $Addr[i]=AR[i]+IR[i]+cin$

位反序加法:  $Addr_r[i]=AR[i]+IR[i]+crin$

其中,  $cin$  为由低位向高位的进位,  $crin$  为由高位向低位的进位。根据两种运算的特点,设计了一种共享加法器的运算电路,如图 4 所示:这样首先求出  $AR$  与  $IR$  的半加结果,之后半加和分别与  $cin$  与  $crin$  进行加法操作,从而输出  $Addr$  与  $Addr_r$  结果。

这样设计的好处是只要基地址和变址准备好,那么顺序寻址和倒序寻址的数据地址将同时产生,同时这种逻辑结构由于共享加法器将有效地减小芯片面积和功耗。

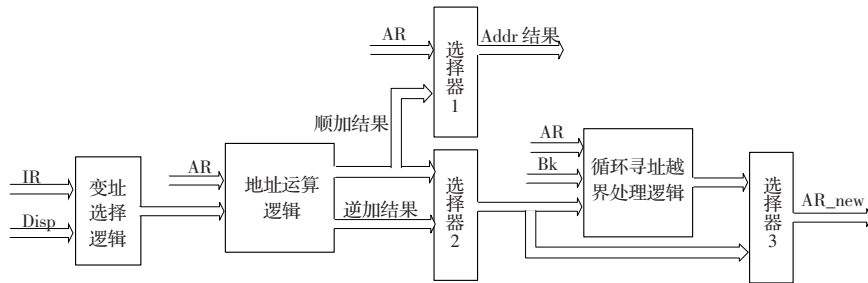


图3 地址产生器的逻辑结构

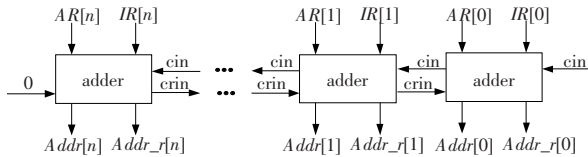


图4 共享加法器的顺序和倒序地址产生逻辑结构

关于循环寻址越界处理逻辑,它的处理算法的流程见图5。

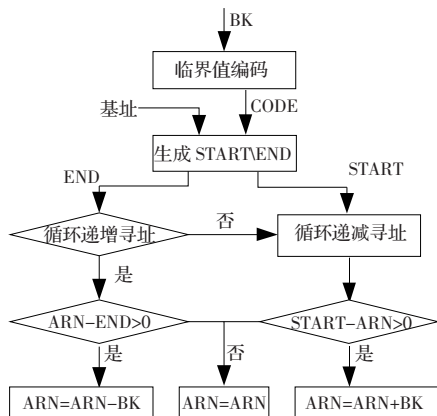


图5 循环寻址越界处理逻辑流程图

循环寻址规则:

(1)要求首地址即  $START$  要满足以下要求:如果  $BK=K$ ; 则  $START$  低位必须为  $m$  个零,且满足  $2^m > k$ 。

(2)步长  $STEP$  要小于循环缓冲区  $BK$  的值,即  $STEP < BK$ 。

临界值编码算法:根据循环缓冲区  $BK$  (二进制表示)大小,从高位向低位寻找第一个“1”,此值便为临界值,之后把最高位至临界值前一位置成1,把临界值至最低位置成0。

$START$  值由基址和  $CODE$  进行位与操作(&)来确定。因为  $CODE$  中,如果有  $n$  个零,且  $BK=K$ ,则必然有  $2^n > k$ ,而  $START$  值最低位零的个数  $m$  必然满足  $m \geq n$ ,则必然有  $2^m \geq 2^n > k$ 。如此便满足了规则1。

$END$  值为  $START$  值与  $BK$  值相加。逻辑上可以用  $START$  和  $BK$  进行位或操作(|)来实现。设  $BK=K$ ,  $START$  值最低位零的个数为  $m$ ,  $BK$  最高位1的位置(从低向高)为  $n$ ,由前面已知  $m \geq n \ln N - 1$ ,当  $BK$  与  $START$  相加时,必然不会存在进位,由此可以简单地用  $BK|START$  来表示  $END$ 。

例如: $BK=16'b0000\_0000\_1010\_0000$

基址= $24'b0000\_1000\_0111\_0000\_1100\_0000$ ,

则有: $CODE=1111\_1111\_0000\_0000$

$START=CODE \& 基址=24'b0000\_1000\_0111\_0000\_0000\_0000$

$END=START|BK=24'b0000\_1000\_0111\_0000\_1010\_0000$

由此便确定了  $START$  与  $END$  值。

## 5 地址产生器的体系结构及性能分析

### 5.1 地址产生器的体系结构

该地址产生器应用在哈佛总线结构的 DSP 芯片中,它是相对于冯·诺依曼结构的并行体系结构。其最大特点是计算机具有独立的存储空间和程序空间,每个空间独立编址,有独立的多条总线,CPU 可以同时进行数据访问和指令读写,提高了数据的吞吐率,从而提高系统运行效率。该地址产生器的体系结构如图6所示:它由两个地址运算的单元 ARAU0 和 ARAU1 组成,处理器内部采用了两条数据地址线,当处理器执行并行指令或者三操作数指令时,两个地址运算单元并行工作,同时在8个辅助寄存器(AR0-AR7)选择基地址,在索引寄存器(IR)中选择变址,在块寄存器(BK)中选择循环缓冲区长度。最后,两个地址运算单元同时输出。

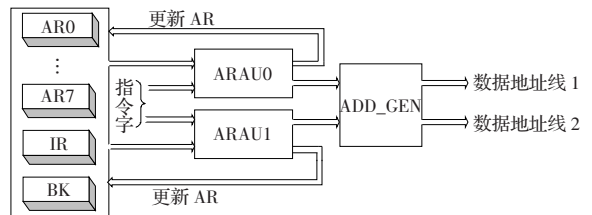


图6 地址产生器的体系结构

由此可以看出,该地址产生器支持多数据流的寻址,能够在—个指令周期完成并行指令,从而在很大程度上提高了处理器的性能。

### 5.2 地址产生器性能分析

假如没有设定专门的地址产生单元,而是用微处理器的核心部件 ALU (算术逻辑运算单元) 进行数据寻址,那么数据运算与地址产生将会串行进行,这样就会制约处理器的速度,从而使 DSP 芯片难以达到较高的硬件处理能力。以带四级流水线的 DSP 处理器为例,四级流水为取指级、译码级、读数级、执行级,每级的执行时间为  $\Delta t$ 。这里连续执行  $n$  条逻辑运算或算术运算指令,其中每条指令都要用到存储器操作数,并且寻址方式为间接寻址,并且设第6条指令为跳转指令,需要用到 PC 相对寻址。流水线的逻辑结构与时空图如图7所示。

通过流水线的时空图,计算出使用 ALU 以及使用 AG 寻址的吞吐率( $TP$ ),加速比( $S$ ),效率( $E$ ),如表3所示。

由此可见,使用独立的 AG 使处理器的流水线效率提高了1倍左右,这样大大减少了指令周期。同时,由于本地地址产生器支持双通路地址的并行计算,这样能够在—个周期内完成两次存储器寻址,有效地支持了并行指令、三操作数指令的执行,对处理器性能的提高起到了至关重要的作用。

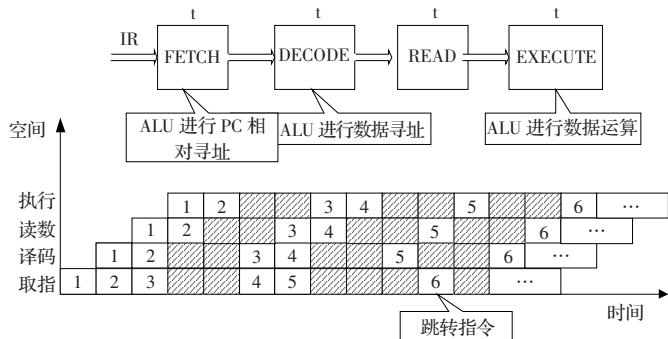


图 7(a)ALU 产生操作数地址时流水线的逻辑结构与时空图

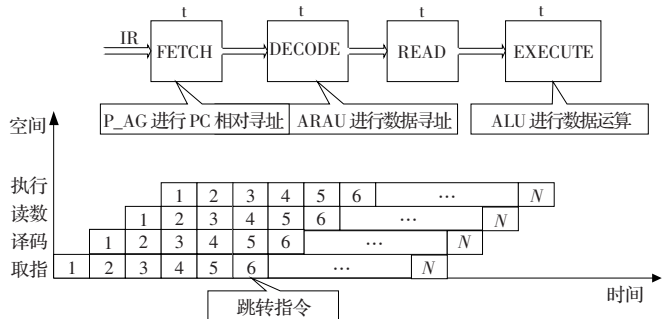


图 7(b)AG 产生操作数地址的流水线结构与时空图

6 结束语

如今微电子技术发展迅猛,工艺尺寸不断减小,使得面积和功耗都不断降低,通过增加硬件模块提高微处理器速度成为开发高性能 DSP 处理器的主要途径之一。提出的地址产生器,

表 3 使用 ALU 寻址与使用 AG 寻址的流水线性能对比

ALU 产生操作数地址的流水线		AG 产生操作数地址的流水线	
<i>TP</i>	$TP_{max(n \rightarrow \infty)}$	<i>TP</i>	$TP_{max(n \rightarrow \infty)}$
$\leq \frac{n}{(1+2n)\Delta t}$	$\frac{1}{2\Delta t}$	$\frac{n}{(n+3)\Delta t}$	$\frac{1}{\Delta t}$
<i>S</i>	$S_{max(n \rightarrow \infty)}$	<i>S</i>	$S_{max(n \rightarrow \infty)}$
$\leq \frac{4n}{1+2n}$	2	$\frac{4n}{k+n-1}$	4
<i>E</i>	$E_{max(n \rightarrow \infty)}$	<i>E</i>	$E_{max(n \rightarrow \infty)}$
$\leq \frac{n}{1+2n}$	$\frac{1}{2}$	$\frac{n}{n+3}$	1

基于哈佛总线结构的处理器,能够完成两条数据地址线的并行操作。该地址产生器不仅支持传统地址产生器的顺序类寻址,而且还支持基于特殊算法的位翻转寻址和循环类寻址。这些特性有效地提高了处理器的性能。

参考文献:

- [1] Sudarsanam A,Liao S,Devadas S.Analysis and evaluation of address arithmetic capabilities in custom DSP architectures [C]//Design Automation Conference,Anaheim,USA,1997,287-292.
- [2] Leupers R,Marwedel P.Algorithms for address assignment in DSP code generation[C]//Proceedings of the IEEE International Conference Computer-Aided Design,1996:109-112.
- [3] Ma Y,Wanhammar L.A hardware efficient control of memory addressign for high-performance FFT processors[J].IEEE Trans Signal Processing,2000,48:917-921.
- [4] 郑伟民,汤志忠.计算机系统结构[M].北京:清华大学出版社,1998.
- [5] 姚天任,江太辉.数字信号处理[M].武汉:华中科技大学出版社,1994.

(上接 77 页)

参考文献:

- [1] Chang M,Tekalp A M,Erdem A T.Blur identification using the bispectrum[J].IEEE Trans Acoust,Speech,Signal Processing,1991,39: 2323-2325.
- [2] Mayntz C,Aach T,Kunz D.Blur identification using a spectral inertial tensor and spectral zeros [C]//IEEE International Conference of Image Processing(ICIP),1999:885.
- [3] Li Qiang,Yoshida Y,Parameter estimation and restoration for motion blurred images [J].IEICE Trans Fundamentals,1997,E80-A (8):1430-1437.

(上接 81 页)

- [15] 邝航宇,金晶,苏勇.自适应遗传算法交叉变异算子的改进[J].计算机工程与应用,2006,42(12):93-96.
- [16] 王小平,曹立明.遗传算法——理论、应用与软件实现[M].西安:西安交通大学出版社,2002.

- [4] Moghaddam M E,Jamzad M.Finding point spread function of motion blur using Radon Transform and modeling the motion length [D].Tehran,Iran:Department of Computer Engineering,Sharif University of Technology,2004.
- [5] Moghaddam M E,Jamzad M.Blur identification in noisy images using radon transform and power spectrum modeling[C]//IEEE 12th International Workshop on Systems,Signal and Image Processing (IWSSIP),Greece,Chalkida,2005:347-352.
- [6] Cuadra D,P Master A,Sapp C.Efficient pitch detection techniques for interactive music [C]//Proceedings of ICMC 2001,La Habana, Cuba,2001:301-307.

- [17] Michalewicz Z.Genetic Algorithms+Data Structure=Evolution Programs[M].北京:科学出版社,2000.
- [18] 李敏强,寇纪松.遗传算法的基本理论与应用[M].北京:科学出版社,2002.
- [19] 周明,孙树栋.遗传算法原理及应用[M].北京:国防工业出版社,1999.