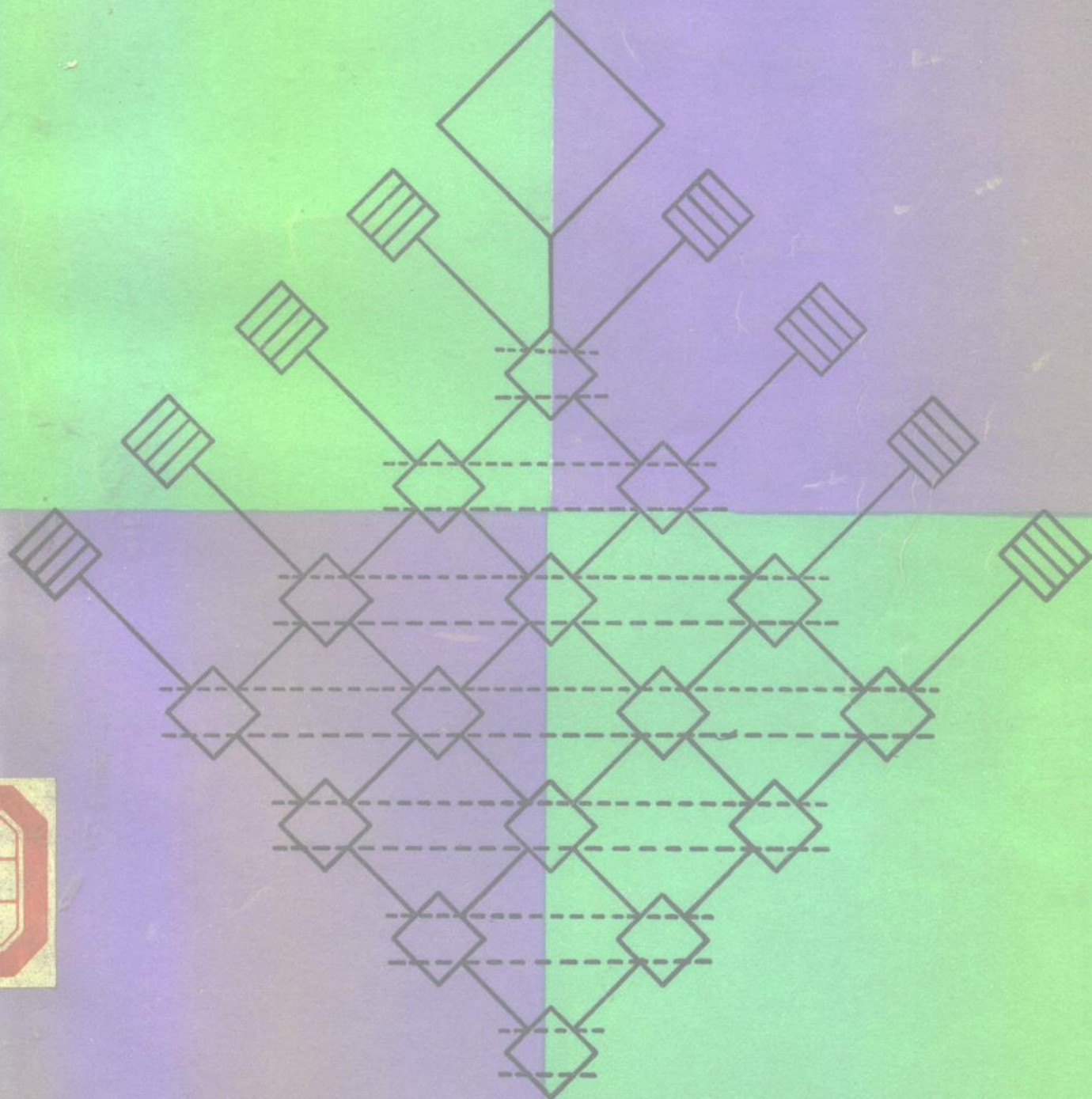


# 实时系统构成技术

沈兰荪 编著

中国科学技术大学出版社



# 实时系统构成技术

沈兰荪 编著

国家自然科学基金资助项目  
高等学校博士学科点专项科研基金资助课题

中国科学技术大学出版社

1993·合肥

## 内 容 简 介

本书是一本关于实时信号处理系统设计的著作。系统介绍了现代实时系统构成技术的发展背景,构成方法,并行算法。给出了一些系统构成的实例。重点讨论了并行处理的基本概念,以及流水线处理、阵列处理、多处理机系统等并行处理的主要方法,对于日益受到重视的 VLSI 并行阵列,以及相应的 VLSI 并行算法也作了重点阐述。对于目前已得到广泛应用的主/从系统也在 DSP 芯片的基础上进行了讨论,并介绍了其在数字图象处理系统中的应用。本书还讨论了分散处理技术,高可用率技术,以及巨大系统的设计问题。

本书可作为电子工程、仪器仪表、自动控制、计算机等专业的研究生或高年级学生的教学用书,也可作为有关专业的科研工作者和工程技术人员的参考用书。

(皖)新登字 08 号

DS79/26

## 实 时 系 统 构 成 技 术

沈兰荪 编著

\*

中国科学技术大学出版社

(安徽省合肥市金寨路 96 号 230026)

中国科学技术大学印刷厂印刷

安徽省新华书店发行

\*

开本: 787×1092/16 印张: 12.25 字数: 298 千

1993 年 2 月第 1 版 1993 年 2 月第 1 次印刷

印数: 1-3000

ISBN 7-312-00421-0/TP·50 定价: 7.80 元

00001111

## 前 言

60年代初期,中国科学院电子学研究所陈宗鹭先生即为中国科学技术大学无线电电子学系讲授了“电子系统”课程。课程通过分析一些典型的电子系统,帮助学生建立有关的系统概念。十年动乱以后,这门课程停开了。1988年根据中国科学院学部委员、中国科大兼职教授陈芳允先生等的建议,学校决定重新开设这门课程,并定名为“实时系统构成技术”,本书即是为这门课程编写的一本教材。

众所周知,近十余年来,微电子技术取得了突飞猛进的发展,从过去的小规模、中规模组件,发展到大规模、超大规模与硅圆片级集成电路 WSI。预计 90 年代,集成电路集成度将每一年半至二年提高一倍,到 2000 年,可能推进到 0.15 微米或 0.1 微米线宽,一块芯片将至少能容纳 5 千万至 1 亿个晶体管。甚至可能达到 10 亿个晶体管的极限。过去的一个机箱、甚或一个机柜的信号处理系统,完全可以集成在单个芯片上,系统设计将从 PCB 板的设计过渡到 VLSI 与 ULSI 芯片的设计。

一些现代电子系统,如战略相控雷达,星载合成孔径雷达等都面临高速、大容量数据流的运算与处理问题。例如,作战管理/指挥、控制、通信(BM/C<sup>3</sup>)系统的信号处理速率要求达到几十亿次/秒,即  $10^9$  次/秒量级;逆合成孔径雷达由于分辨率很高,有关的运动补偿相当复杂,实时运算速度要求高达千亿次/秒,即  $10^{11}$  次/秒量级。显然,高速大容量的数据流,以及对信号处理实时性的要求已是现代信号处理系统的一个主要特点。

目前,进一步提高处理芯片的运行速度受到了一些限制,难以满足现代系统设计的要求,而传统的 Von-Neumann 体系结构的经典有限态机制也限制了构成系统的处理速度。显然,现代实时系统构成技术必须采取新的观点与方法。当今最值得重视的一个研究方向是发展由大量的基本处理单元构成的并行处理器阵列,脉动阵列与波前阵列是这方面的突出代表。它们的模块化、互连规则性与就地互连等结构特点,很适应 VLSI 的实现,其对数据的高度并行流水处理能力使其具有极高的处理效率,从而使实时处理技术的发展迈上了一个新台阶。

本书主要介绍用于信号处理的现代实时系统构成技术,是以作者在中国科学技术大学讲授这门课程的讲稿为基础编写的。全书共分 15 章,讨论了实时系统构成技术的发展背景,基本概念,构成方法,VLSI 并行算法,以及若干系统构成的实例。由于 VLSI 系统集成设计以及相应的算法讨论中国科学技术大学另设有课程讲授,本书未作深入讨论。

本书可作为电子工程、仪器仪表、自动控制、计算机等有关专业的研究生或高年级学生的教学用书,也可作为有关专业的科研工作者和工程技术人员的参考用书。

作者特别感谢陈芳允、陈宗鹭、柴振明等先生的鼓励与帮助,作者的学生谢卫国同学帮助整理了第六章的部分材料,李天福同学整理了表 6-7,胡立红、沈晓军同志帮助抄写了全部书稿,李智群、仲敏同志帮助校对了部分书稿,在此一并致谢。

本书的出版得到了《国家自然科学基金资助项目》和《高等学校博士学科点专项科研基金资助课题》的赞助。

沈兰荪

1992 年 9 月于中国科学技术大学电子三楼

# 目 次

|                                       |      |
|---------------------------------------|------|
| 前 言 .....                             | (II) |
| 第一章 引论 .....                          | (1)  |
| 1-1 对实时信号处理技术的要求 .....                | (1)  |
| 1-2 微电子技术的进步 .....                    | (2)  |
| 1-3 并行处理 .....                        | (5)  |
| 第二章 信号处理 .....                        | (7)  |
| 2-1 概述 .....                          | (7)  |
| 2-2 FFT 在数字信号处理中的重要地位 .....           | (8)  |
| 2-3 FFT 的基本原理 .....                   | (11) |
| 2-4 FFT 的实现 .....                     | (15) |
| 2-5 讨论 .....                          | (18) |
| 第三章 专用集成电路 .....                      | (19) |
| 3-1 概述 .....                          | (19) |
| 3-2 ASIC 芯片 .....                     | (20) |
| 3-3 工艺技术 .....                        | (25) |
| 3-4 圆片规模集成与三维集成技术 .....               | (26) |
| 第四章 并行处理的基本概念 .....                   | (27) |
| 4-1 概述 .....                          | (27) |
| 4-2 并行处理的定义 .....                     | (28) |
| 4-3 流水线处理 .....                       | (31) |
| 4-4 阵列处理 .....                        | (33) |
| 4-5 多处理机系统 .....                      | (34) |
| 4-6 大规模并行 .....                       | (36) |
| 4-7 关于 RISC 的简要讨论 .....               | (36) |
| 第五章 DSP 芯片 .....                      | (38) |
| 5-1 概述 .....                          | (38) |
| 5-2 TMS320 系列 DSP 芯片 .....            | (44) |
| 5-3 DSP 芯片的结构特点 .....                 | (47) |
| 5-4 基于 DSP 芯片的系统构成 .....              | (49) |
| 5-5 TMS320C40——一种用于并行处理的 DSP 芯片 ..... | (52) |
| 第六章 主从式系统的构成 .....                    | (55) |
| 6-1 概述 .....                          | (55) |
| 6-2 总线技术 .....                        | (56) |
| 6-3 双端口存储器的设计 .....                   | (59) |

|             |                             |              |
|-------------|-----------------------------|--------------|
| 6-4         | PC 环境下的 DSP 系统 .....        | (63)         |
| 6-5         | PC 环境下 TMS320C30 模板设计 ..... | (64)         |
| 6-6         | 模块式信号处理系统 .....             | (67)         |
| 6-7         | 采用 TMS320C30 模板实现 FFT ..... | (70)         |
| <b>第七章</b>  | <b>流水线处理 .....</b>          | <b>(82)</b>  |
| 7-1         | 概述 .....                    | (82)         |
| 7-2         | 线性流水线原理 .....               | (83)         |
| 7-3         | 通用流水线与预约表 .....             | (85)         |
| 7-4         | 最大时间差流水线 .....              | (87)         |
| <b>第八章</b>  | <b>多处理机系统结构 .....</b>       | <b>(89)</b>  |
| 8-1         | 概述 .....                    | (89)         |
| 8-2         | 松散耦合的多处理机系统 .....           | (90)         |
| 8-3         | 紧密耦合的多处理机系统 .....           | (92)         |
| 8-4         | 时分总线 .....                  | (94)         |
| 8-5         | 交叉开关与多端口存储器 .....           | (96)         |
| 8-6         | 多级互连网 .....                 | (97)         |
| 8-7         | 讨论 .....                    | (103)        |
| <b>第九章</b>  | <b>阵列处理 .....</b>           | <b>(106)</b> |
| 9-1         | 概述 .....                    | (106)        |
| 9-2         | 阵列处理机的结构 .....              | (106)        |
| 9-3         | 互连网技术 .....                 | (108)        |
| 9-4         | 关联处理的基本概念 .....             | (110)        |
| 9-5         | 讨论 .....                    | (114)        |
| <b>第十章</b>  | <b>VLSI 并行阵列 .....</b>      | <b>(119)</b> |
| 10-1        | 概述 .....                    | (119)        |
| 10-2        | 脉动阵列 .....                  | (121)        |
| 10-3        | 波前阵列 .....                  | (123)        |
| 10-4        | 可编程序的处理器阵列结构 .....          | (125)        |
| 10-5        | 讨论 .....                    | (127)        |
| <b>第十一章</b> | <b>VLSI 并行算法 .....</b>      | <b>(129)</b> |
| 11-1        | 概述 .....                    | (129)        |
| 11-2        | 矩阵乘法 .....                  | (129)        |
| 11-3        | 卷积 .....                    | (131)        |
| 11-4        | 数字滤波 .....                  | (137)        |
| 11-5        | FFT .....                   | (141)        |
| <b>第十二章</b> | <b>分散处理技术 .....</b>         | <b>(146)</b> |
| 12-1        | 概述 .....                    | (146)        |
| 12-2        | 分布式系统 .....                 | (147)        |
| 12-3        | 分散处理技术 .....                | (148)        |

|                            |              |
|----------------------------|--------------|
| 12-4 地区分散型处理系统 .....       | (150)        |
| 12-5 分布式数据库 .....          | (153)        |
| 12-6 讨论 .....              | (154)        |
| <b>第十三章 数字图象处理系统 .....</b> | <b>(156)</b> |
| 13-1 概述 .....              | (156)        |
| 13-2 技术指标 .....            | (156)        |
| 13-3 伪彩色处理 .....           | (157)        |
| 13-4 图象处理系统的构成 .....       | (161)        |
| 13-5 应用软件 .....            | (166)        |
| 13-6 一个微机数字图象处理系统 .....    | (169)        |
| <b>第十四章 巨大系统设计举例 .....</b> | <b>(173)</b> |
| 14-1 概述 .....              | (173)        |
| 14-2 灾害测报监控系统建设的指导思想 ..... | (173)        |
| 14-3 灾害测报监控系统的组成 .....     | (174)        |
| 14-4 关键技术研究 .....          | (179)        |
| 14-5 系统建设的初步设想 .....       | (179)        |
| <b>第十五章 系统的高可用率 .....</b>  | <b>(180)</b> |
| 15-1 概述 .....              | (180)        |
| 15-2 故障源的分析 .....          | (181)        |
| 15-3 电磁兼容性设计 .....         | (182)        |
| 15-4 可靠性设计 .....           | (183)        |
| 15-5 容错技术 .....            | (185)        |
| <b>参考文献 .....</b>          | <b>(188)</b> |

# 第一章 引 论

## 1-1 对实时信号处理技术的要求

信号处理(signal processing)<sup>[2,3,4,5,6,8]</sup>, 是利用模拟、数字或光学的方法分析或处理由时变物理过程所产生的物理信号的科学。随着信号处理算法执行成本的不断下降, 信号处理的应用范围也越来越广。除了传统的研究内容——频谱分析与滤波器设计之外, 在信号的结构、逼近、内插、外推; 利用部分信息恢复信号; 高分辨率的谱分析; 高效自适应滤波等方面新的成果不断涌现, 多变量信号处理技术也愈来愈受到人们的重视。信息处理是一门正在发展的学科, 它的范畴很广, 其主要内容为: 信号在时间域、频率域或其他特征域上的性质; 信号经过各种系统后的存在形式; 设计各种处理系统以期产生具有给定性质的信号, 或得到按某种准则的最优逼近信号; 随机信号与噪音的统计特性, 估计信号的特征参数, 以及从噪音中提取信号等等。

经典信号处理方法主要是利用信号在频率域的一些特征进行滤波处理, 主要的处理手段是采用集中参数元件构成各种模拟滤波器 (LC 滤波器与 RC 有源滤波器)。这都是无限脉冲响应滤波器。从现代数字信号处理观点来看, 对于  $N$  点输入数据, 进行与  $N$  同一数量级的运算处理后输出  $N$  点数据, 即吞吐速率为  $O(N)$  次。

自 50 年代后期发展起来的数字信号处理技术, 是以采样的离散信号为处理对象。在时域上发展了离散 FIR(finite impulse response)和 IIR(infinite impulse response)算法, 相关处理技术, DFT(discrete fourier transform)等变换, 以及梯度型的自适应滤波算法、线性预估等方法。这些算法对吞吐速率的要求是  $O(N^2)$  次。60 年代数字计算技术的水平是不能满足这些算法对实时(real time)处理的要求的。

1965 年 Cooley-Turkey 的快速傅里叶变换 FFT(fast fourier transform)算法的出现, 将 DFT 算法要求的吞吐速率降为  $O(N\log_2 N)$ 。这样, 70 年代出现了各类基于 FFT 算法的实时信号处理硬件, 如高速位片式微处理器, 高速并行乘法器等。

与此同时也发展了一些离散模拟信号处理器件, 如 CCD 器件, 新型表面声波器件 SAW 等。这类器件保留了模拟的信号采样值, 但在时间上却是离散的, 而且具有体积小、重量轻、频带宽、便于实时处理等特点。用这类器件可以构成 Chirp (线性调频脉冲压缩) 变换设备, 模拟串行滤波器等。这类实时信号处理硬件是专用器件, 要改变参数或算法很不方便。

80 年代初出现的将微处理器与快速并行乘法器集成在一块 VLSI 中的单片信号处理器——DSP(digital signal processor)芯片。这类芯片采用了修正的 Harvard 结构, 其算法与逻辑控制由软件实现, 从而为实时信号处理提供了廉价而有效的工具。单片 DSP 芯片可提供  $10^6 \sim 10^7$  次/秒的吞吐速率, 采用多片 DSP 的组合系统, 吞吐速率可达到  $10^7 \sim 10^8$  次/秒以上。

现代信号处理算法的研究往往是基于更实用的非平稳的噪音模型, 并利用信号结构的

先验信息,这样可以大大提高处理效果,但要求的计算量也相应增大许多。用直接最小二乘技术做自适应噪声抵消,可以更快收敛,且此梯度法有更好的跟踪性能,但这种算法要做矩阵的正交三角化,算法比较复杂。超指向性阵处理技术,采用信号的空间多点源模型可以改进分辨率,但要求在每个频率成分上求解特征系统或广义特征系统。这些算法的共同特点是大量采用线性代数运算,主要是矩阵运算,包括矩阵与矩阵相乘,矩阵求逆,特征值特征矢量分解,求解线性方程组等。这些算法要求的吞吐量为 $O(N^3)$ 。在实时处理时通常要求吞吐速率高达 $10^9$ 次/秒或更高。

一些现代电子系统,如战略相控雷达、星载合成孔径雷达(SAR)等都涉及高速、大容量数据流的运算处理问题。例如,作战管理/指挥、控制、通信(BM/C<sup>3</sup>)系统的信号处理速率要求达到几十亿次/秒( $10^9$ 次/秒)。SAR信号处理实质是二维(斜距与横向距离)的相关运算。为得到实用的SAR图象,还需进行距离迁移校正、多普勒中心频率与频偏估计,以及多态处理等。对相关运算,采用基2FFT变换,获得一个输出样本需进行 $L=4+2\log_2 N$ 次实乘运算。取 $N=2048$ ,则 $L=26$ 。对40MHz输入信号采样率,完成二维相关约需 $2.08\times 10\text{MOPS}$ (20.8亿次/秒)实数乘法运算。对于双频多极化星载SAR,输入信号样本速率达到1000兆/秒量级,运算速度要求达到约500亿次/秒( $5\times 10^{10}$ 次/秒)实数乘法运算。逆合成孔径雷达(ISAR)由于分辨率更高,运动补偿更为复杂,实时运算速度要求千亿次/秒( $10^{11}$ 次/秒)以上。

显然,高速大容量的数据流以及对信号处理实时性的要求,是现代信号处理系统的一个主要特点。

光学信号处理是最早采用的一种信号处理技术,但一直未能得到广泛的应用。1960年激光器的出现推动了光学信号处理的发展。从理论上说,光学处理系统的速度仅受光速的限制,实际上处理速度仅受系统准备输入数据的限制。今天在侧视综合孔径雷达,无模糊光学成像与遥感数据分析等一些领域已成功地采用了光学信号处理技术。随着光计算机的发展,光学信号处理技术越来越受到人们的重视。

应该指出的是,尽管国外正在广泛研究光学信号处理技术,但由于数字处理的高度灵活性、可编程和可重组装等优点,在今后相当长时间内,数字处理仍是解决现代信号处理问题的主要手段。其特点表现为信号处理理论、算法、结构和实现技术的紧密结合。而具体技术途径则是并行处理技术和VLSI技术的结合。

## 1-2 微电子技术的进步

近十年来,微电子技术取得了突飞猛进的发展,从小规模组件、中规模、大规模集成电路发展到超大规模与硅晶片级集成电路WSI(wafer scale integration)。预计90年代,集成电路的发展,每一年半至二年其芯片的集成度会提高一倍。半导体技术的成熟,1989年为1微米和0.8微米线宽,到1994年可能为0.35微米线宽,2000年可能推进到0.15微米或0.1微米线宽,届时一块集成电路内至少容纳5千万至1亿个晶体管,甚至可能达到10亿个晶体管的极限,是目前集成度的50倍以上。过去的一个机箱,甚至一个机柜的信号处理系统,完全可以集成在单个芯片上。系统设计就从PCB板(printed circuit board)

的设计过渡到 VLSI 与 ULSI 芯片的设计。

1992 年初, 日本日立制作所采用 0.3 微米的微细加工技术, 将 102 万个半导体器件集成在一块 8 毫米见方的芯片上, 研制成一种每秒钟能处理 10 亿条指令的微处理机。芯片集成时采用了双极型与 CMOS 两种工艺, 在芯片上设置了可以并行操作的两个处理器、存储器与四个运算器, 处理机的运算速度达到每秒钟 2.5 亿次。这是目前世界上最快的微处理机。

在 2000 年前, 微电子技术可望取得下述进展:

- 千兆赫速度的信号处理芯片;
- 采样速度为千兆次/秒的 ADC 转换技术;
- 亚微米特征尺寸;
- 圆片规模的逻辑和存储器的集成技术;

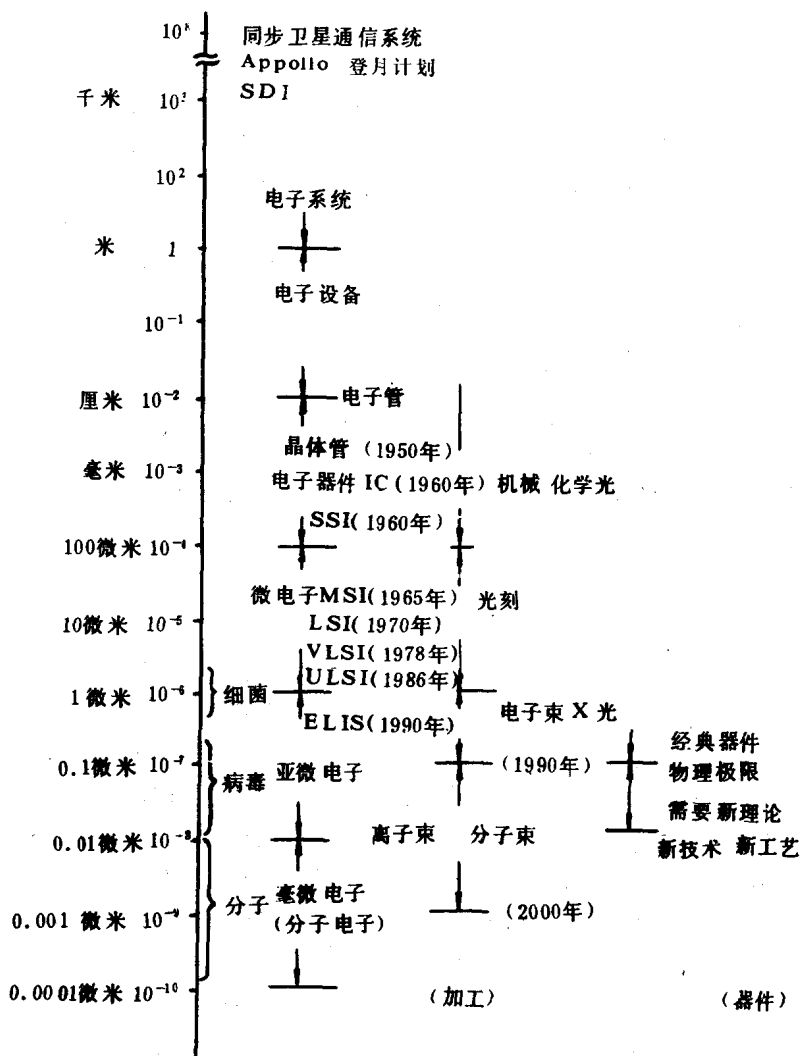


图 1-1 电子器件的发展图

- 三维集成技术;
- 千兆赫速度 GaAs 集成电路。

集成度进一步发展有无限制? 随着集成度的提高, 半导体芯片的加工线宽将接近半导体器件电子运动的一些特征长度, 如自由程、电子波长等。集成度提高的另一个限制, 在于是否能制造出所需线宽与精度的工艺设备, 这主要决定于图形的产生与刻蚀。一般认为光刻的理论极限为 0.5 微米, 电子束刻蚀的可用极限为 0.3 微米, 离子束可做到更细的线宽。可以认为, 利用整体效应所设计的微电子器件的极限, 本世纪末之前有可能达到, 见图 1-1。

目前, 单个 GaAs 三极管的开关速度已经达到 10 微微秒, 相应器件的尺寸可以做到 0.1 微米左右。但是要在单位芯片面积上做出更多的 0.1 微米尺寸的器件, 却遇到了困难, 主要有:

### 1. 带限效应

首先是芯片与芯片之间互连以及芯片上、下的连线造成的带限效应。

随着 VLSI 集成度的提高, 假定芯片上所有元件空间均缩小 2 倍, 则晶体管的开关时间相应地缩小 2 倍, 但是因连线截面和间距的减小而产生的电阻虽然减小 2 倍, 而分布电容却增大 2 倍, 使得连线元件之间电连接的时间常数  $RC$  保持不变。因而最终导致信号的传输速度主要决定于互连延迟, 从而限制芯片的运算速度。如果考虑到快速材料 GaAs 技术的发展和电路的复杂程度在不断增加, 导致门电路之间的连接并不随之缩短, 而是大致保持一定值, 则传输速度受电连接的影响将会大大加剧。

其次, 随着 VLSI 集成度的提高, 芯片所需要的外引脚数也将随之增加, 按照 Rent 定则, 所需外连的引脚数  $M$  与门器件数  $N$  之间的关系为:  $M \approx N^{0.67}$ 。而芯片周边能够提供的引脚数  $M \approx N^{0.5}$ 。因此随着门集成数目的增大, 芯片将没有足够的引脚供信号输出。

另一方面, 集成度和运算速度的增大, 又导致了传输线相互之间电磁干扰的加剧, 以至芯片无法进行工作。

### 2. 时钟分布问题

VLSI 芯片中的同步运行是由传送同步信号到芯片的各个部分来实现的, 即所谓的时钟信号。随着运算速度的提高, 在时钟分布电路的设计中遇到两方面的困难。首先时钟脉冲信号由时钟驱动器到系统的各个部分的传输线基本上以平行方式连接。连线的分布电容作为时钟驱动器的负载电容加在电路上, 而时钟驱动器输出电压的传递时间正比于负载电容与驱动器固有电容之比。这样连线电容引起信号脉冲增宽; 其次, 传统的电连接造成了系统的不同部分在不同时刻受到相同状态的时钟信号作用, 即造成了时钟扭曲问题。时钟分布要求增大金属布线, 这样就增大了芯片表面的占有量, 限制了 VLSI 的集成度。

对于一般的金属导体来说, 光速大约是 4 纳秒/米。若时钟速度为 100 兆赫, 则数据的传输时间间隔为 10 纳秒。在此期间内, 反射波可以传播 2.5 米。显然, 如果系统中的互连线长度小于 20~30 厘米, 则在 10 纳秒的时间内传输线上的反射波必定会消失。假如时钟速度提高到 500 兆赫, 则数据传输时间间隔缩短为 2 纳秒。在 2 纳秒的时间里, 反射

波的行程只有 50 厘米。因此，如果往返的长度是 40~60 厘米，则一个数据的第一次或第二次反射波就会出现在下一个数据期间。如果反射波出现在时钟信号的上升沿，就可能引起读数据错误。根据目前对数据信号上升时间的控制经验与水平来看，时钟速度可以高达 100 兆赫（10 纳秒周期），但时钟速度难以进一步提高到 1 吉赫（1 纳秒周期）。

总线结构带来了另一个问题。总线上通常接有若干个驱动器/接收器，它们共用一对连线。这条总线的两端各接一个端电阻  $Z_0$ ，以便消除或衰减掉反射波。由于这种结构可以方便地共用传输线，而且可以有效采用广播通信方式，因而它是颇受欢迎的互连形式。

但是，总线上的每一个接头实际上是物理上的不连续点，当电压/电流波经过接头时就会产生反射。理想的接头应当长度为 0、阻抗为无穷大；但实际接头具有一定的长度，其阻抗虽高但不是无穷大，而且还存在杂散电容。这些因素凑在一起，引起的反射波将使总线结构不适合于高速系统。

根据一般的规律，总线适合于 100 兆赫以下；当时钟速度超过 400 兆赫时，就必须采取点至点的连接方法；在 100~400 兆赫的范围内若要使用总线，就必须非常仔细地设计总线的几何结构和选择器件。

现代通用数字计算机是按 Von-Neumann 的经典有限态机制进行运转，即采用按时序串行运行的方式，从而限制了它的运算速度。人们在研究计算机的并行运行方式时，也要遇到一个“瓶颈”问题。“瓶颈”主要表现在中央处理单元（CPU）与存储器之间。CPU 的每个输出必须与相关存储器单元相连，反之所有存储单元都必须同时能与 CPU 相连，形成扇出扇入。计算表明，为了不引起电磁串扰，采用金属导线连接方式，每平方厘米能通过的导线数不应超过  $10^3$  根。

显然，提高构成实时信号处理的处理单元的速度尚受到一定限制。目前最值得重视的一个研究方向是发展由大量的基本处理单元构成的数据级并行的处理器阵列。随着微电子学 VLSI 技术的发展，目前已有可能在一个 VLSI 芯片上集成由大量处理单元组成的阵列。VLSI 设计要求各个处理单元尽量一致，规则排列，在单元间只安排相邻单元间的局部通信。这就要求将数字信号处理算法与 VLSI 电路设计结合起来进行。70 年代末 80 年代初出现了满足实时信号处理要求和适合 VLSI 实现的各种并行处理计算结构。美国 Carnegie-Mellon 大学 H.T.Kung 教授提出脉动阵列 (systolic array) 和波前阵列 (wavefront array) 结构是这方面的突出代表。它们的模块化、互连规则性和就地互连 (local interconnection) 等结构特点很适宜于 VLSI 实现，它对数据的高度并行流水处理能力使其具有极高的处理效率。它们的出现使超高速实时处理成为可能。

### 1-3 并行处理

并行处理是通过多机并行工作的方法提高整个处理系统的速度。和大型通用计算机相比，高速并行处理机独有的特点是：它是一种高度流水并行的、专用性很强的处理系统。一般的计算机，特别是微型机，很难胜任实时数据信号处理（DSP）任务。从 DSP 的基本运算可以看出：一是 DSP 基本运算中，有大量的乘法、矢量旋转、以及三角函数运算；二是一般计算机的体系结构，对 DSP 的基本运算，经常出现的循环缓冲数据结构只

能提供繁琐的寻址操作；三是 DSP 运算任务不仅具有数据量大，数据结构规整与局部性，而且处理算法大多具有递推特点，一些 DSP 算法具有很好的并行性，这些在一般计算机系统中不能有效地加以利用。

DSP 系统一般应具有适应 DSP 算法的基本运算指令；具有适应 DSP 数据结构的寻址机构；同时可以充分利用算法中的并行性，以达到实时信号处理的要求。换句话说，DSP 系统是以算法特点为基础来进行设计的，体系结构直接反映了算法的特点。微电子学和 VLSI 技术的高速发展，也为算法与体系结构的匹配提供了实现的前提。VLSI 阵列结构即为解决多机并行处理的关键——处理单元的互连问题提供了条件。

由于并行结构随算法的不同而异，要求根据各种算法设计不同的 VLSI 电路。并行处理是当代计算机的一项新技术，可以分为两大类。第一类为指令级或功能级并行，在这种方式下，并行工作的处理单元数量不太大，一般在几十个以内。第二类为数据级的并行，在这种方式下，被处理的数据被分成若干块，每块由一个处理单元执行。Thinking Machines 公司的通用并行处理机 Connection machines 由 64K 个处理机并行构成，其吞吐速率高达  $10^9 \sim 10^{10}$  次/秒。

现代数字信号处理算法的实用性，往往取决于这一算法的可行性，看其是否适合于并行处理。信号处理超级芯片的设计，即数字信号处理系统的集成，包括了从信号处理算法、并行和流水结构、逻辑合成、电路设计、版图设计等各个方面。传统的基于通用计算机上的信号算法已经过时，而应将算法改造成为适合于 VLSI 的新算法，目前已经形成了一门 VLSI 信号处理算法的新兴学科。在结构上的主要发展趋势是采用多处理机组成并行和流水线的结构，最有代表性的是脉动阵列和波前阵列结构。将算法直接映射成结构的阵列编译器仍处于初期阶段，还未实用化。在逻辑合成，电路设计和版图设计方面，已经出现了大量的 CAD 工具，给设计者提供了大量的帮助和支援，提高了设计的自动化程度，大大减轻了工作量。但这些 CAD 工具，还必须领先设计者的经验和知识，才能用好，达到优化设计。当前微电子标准加工线和 CAD 工具已为逻辑到版图的设计提供了有力的支持。系统集成技术是要解决整个 VLSI 设计的优化，包括从算法直到印刷版图的全面优化，必须前后反复调整和修改设计，其中算法和结构的优化设计更为关键。

图 1-2 表示了现代信号处理系统的设计流程。显然在 VLSI 设计环境中，软件与硬件之间的边界已变得十分模糊。重要的是要进行算法分析，并把算法映射到 VLSI 结构体系上去。为了实现高度并行的处理结构，还要进行很多方面的改进，包括在通信与计算之间的折中，并行机制，阵列结构，编程技术，以及关于数字信号处理算法、数值稳定性分析等。

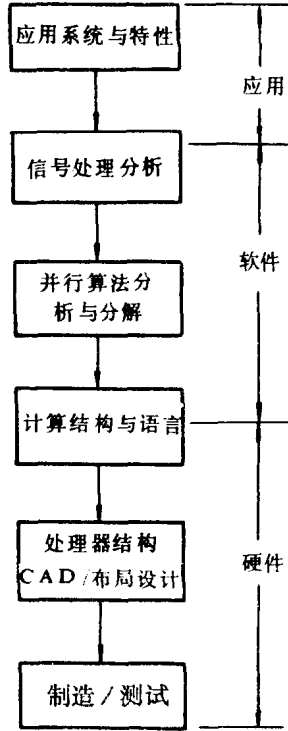


图 1-2 现代信号处理系统设计流程

## 第二章 信号处理

### 2-1 概 述

数字信号处理是研究如何用数字或符号的序列来表示信号,并对这些序列进行运算处理的一门学科。其内容涉及滤波、相关、卷积、变换等各种处理手段。其目的是滤除信号的背景噪音干扰,削弱信号中的多余信息量,以便估计信号的特征参数,或变换成易于分析或识别的形式<sup>[2,3,4,5,6,8,9,12,14]</sup>。

从噪音中提取信号是数字信号处理的主要任务之一。如果噪音  $n(t)$  与信号  $s(t)$  是加法性结合的,即观测值  $x(t) = s(t) + n(t)$ 。当信号与噪音频带基本不重叠或重叠时,可以采用一些线性滤波方法(如平滑、叠加处理等)进行频域滤波。这实际上是采用线性非移变系统对信号进行数字滤波。

线性非移变系统之所以重要,一方面是由于它易于分析与描述,容易导致相当完备而有效的数学表达式;另一方面是可以设计并实现具有各种信号处理功能的线性非移变系统。这样一类加性噪音干扰的成功分离,是由于线性非移变系统符合迭加原理的直接结果。

在实际应用中,经常会遇到非相加性噪音对信号的污染。例如在语音、图象、雷达、声纳、遥感、地震勘探与电生理学等领域中,常常遇到干扰与信号是乘性或卷积组合在一起的情形。此时,观测值分别为  $x(t) = s(t) \cdot n(t)$ ,  $x(t) = s(t) * n(t)$ 。对于这种情况,传统的线性滤波方法将无能为力,而必须借助于非线性系统来进行处理。人们为了把比较成熟的线性系统设计方法应用于这类非线性组合信号的处理,提出了一类服从广义迭加原理的非线性系统。这类系统可以表示为输入与输出矢量空间之间的同态变换,通常称为广义线性滤波或同态滤波。同态系统已成功地应用于处理乘性或卷积组合的信号与噪音。采用 FFT 先将信号与噪音的卷积结合转换成乘法结合,再取对数转化为加法性结合,以实现信号的滤波处理。

40 年代初,由于军事上的需要, N.Wiener 把随机过程与数理统计的观点与方法引入到控制中来,建立了 Wiener 滤波理论。当信号与干扰噪音是具有一定统计特性的平稳随机过程时,只要滤波器的脉冲响应满足 Wiener-Hopf 方程,滤波器的输出与信号的均方误差为最小。50 年代后期, Wiener 滤波器已推广到非平稳与时变情况。Wiener 理论的一个主要缺点,就是 Wiener-Hopf 方程不易求解,对观测值要追溯到无限的过去,实际上是无法实现的。

60 年代, Kalman 把状态空间法引入到滤波中来,他以线性、无偏、最小方差为准则,在时域得到了新的递推滤波算法,即 Kalman 滤波。Kalman 滤波在离散情况下使一阶差分方程组便于计算机处理;采用状态转移矩阵来描述,既可用于线性系统也可用于非线性系统;既可用于时不变系统也可用于时变系统。Kalman 滤波器实质上是把估值理论中的最小二乘法,加权最小二乘法,递推最小二乘法与最小方差法结合起来的递推估值。

实际工程环境，往往要求能够在没有或只有很少的信号统计知识的情况下，对输入观测值作连续递归处理，Widrow 提出的自适应处理即是这样一种处理方法。它不仅有利于适应信号的非平稳性，而且便于实时处理。

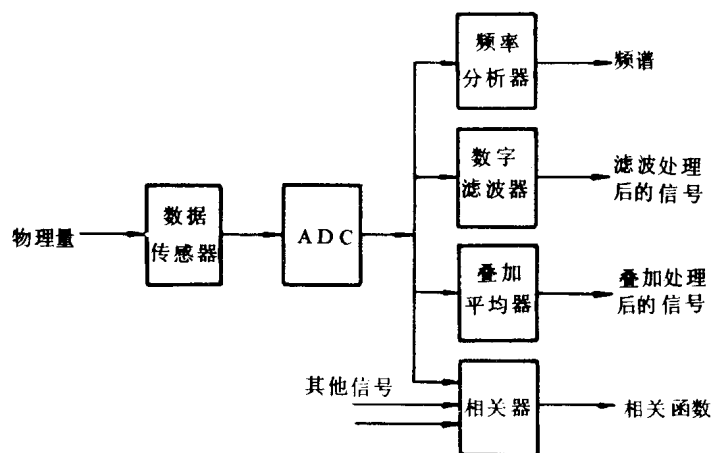


图 2-1 数字信号处理中若干典型的运算

图 2-1 示出了数字信号处理中若干典型的处理运算。一些物理量（例如位移、速度、加速度、强度、压力、温度、颜色等）通过数据传感器(data sensor)变换为电压，再由模数转换器 ADC 转换为数字信号。这些数字信号可以先记录存储下来供进一步处理，称为离线（off-line）处理，也可如图 2-1 那样，在采集的同时进行实时（real time）处理。

数字信号处理常用的一个运算是频谱的计算（即频谱分析）。这是利用待处理的时域信号的快速傅里叶变换来显示这些信号的某些内在特性，以更方便地解决问题。频谱分析通常可以给出信号的幅值、相位、功率和能量对频率的分布。

数字信号处理中另一类常用的运算是数字滤波。这通常用来对原始的或输入的数字信号进行一些运算。实际上，数字滤波是一些算法，它将所给的数字信号（一些以均匀速率出现的数字序列）转化成另一些频谱被修正了的数字信号，这就是众所周知的幅度与相位响应，以及传递函数的零极点的概念。

数字信号处理中又一常用的运算为“叠加平均”（又称为信号平均，signal averaging）。当信号的频谱与噪音的频谱全部或部分相重叠时，一般的滤波技术不再适用，用信号的叠加平均技术则有可能改善信噪比。

第四类运算为“相关”。这是一种把信号与其自身、或与其它信号进行比较的特殊处理，可用来表征时间域中一个信号的不同部分之间或两个信号之间的相似性。

## 2-2 FFT 在数字信号处理中的重要地位

应该强调指出的是，在众多的信号处理领域中，时间—频率域变换是一个重要的组

成部分，而高效的时间——频率域变换的关键是快速傅里叶变换算法 FFT。

FFT 的重要应用之一就是高速卷积。两个信号  $x(n)$  与  $h(n)$  的卷积定义为：

$$x(k) * h(k) = \sum_{n=0}^{N-1} x(k) \cdot h(k-n)$$

根据 DFT 的性质，上式等效为：

$$x(k) * h(k) \Rightarrow X(n) \cdot H(n)$$

如果信号是相当长的函数，直接完成卷积所需的计算时间是很大的，但由上式可知，这种卷积运算相当于在变换域中两个信号的 FFT 的简单相乘。这就使计算时间大大缩短，因为 FFT 所用的时间正比于  $\frac{N}{2} \log_2 N$ ，而直接计算卷积所用时间正比于  $N^2$ 。N 越大，节约的时间越多。这样，卷积计算由下面三个基本步骤组成：① 利用 FFT 计算出两个信号的 DFT。② 在各频率点处，将两信号的变换值相乘。③ 再一次运用 FFT 算法，计算出变换式乘积的反变换。

FFT 的重要应用之二是数字滤波。首先计算出信号  $x(k)$  的 DFT，再在频率域按要求的频率响应对输入频谱进行加权处理，最后，再计算出 DFT 的反变换，求得  $y(k)$ 。此即为经过数字滤波处理后的结果（图 2-2）。

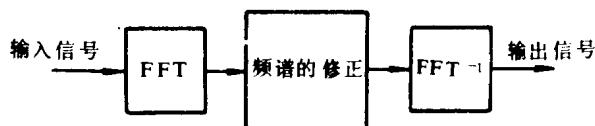


图 2-2 利用 FFT 进行数字滤波

FFT 的重要应用之三是相关处理。

对于已知的两个离散时间信号  $x(k)$  与  $y(k)$ ，其互相关函数  $R_{xy}(m)$  可表示为：

$$\begin{aligned} R_{xy}(m) &= \overline{x(k) \cdot y(k-m)} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} x(k) y(k-m) \end{aligned}$$

显然它是延时  $m$  的函数。

其自相关函数  $R_{xx}(m)$  可表示为：

$$\begin{aligned} R_{xx}(m) &= \overline{x(k) \cdot x(k-m)} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} x(k) x(k-m) \end{aligned}$$

利用 DFT 的滞后乘积性质，由上式可得：

$$\begin{aligned} \text{DFT}[R_{xx}(m)] &= \frac{X(n) \cdot X^*(n)}{N} \\ &= \frac{|X(n)|^2}{N} = S_{xx}(n) \end{aligned}$$

这里  $S_{xx}(n)$  是已知信号  $x(k)$  的离散自功率谱。它的逆变换就是自相关函数  $R_{xx}(m)$ 。

$$\text{DFT}^{-1}[S_{xx}(n)] = R_{xx}(m)$$

离散互功率谱  $S_{xy}(n)$ ，同样可以表示为：

$$\text{DFT}[R_{xy}(m)] = \frac{X(n)Y^*(n)}{N} = S_{xy}(n)$$

互功率谱的逆DFT就是互相关函数，即

$$\text{DFT}^{-1}[S_{xy}(n)] = R_{xy}(m)$$

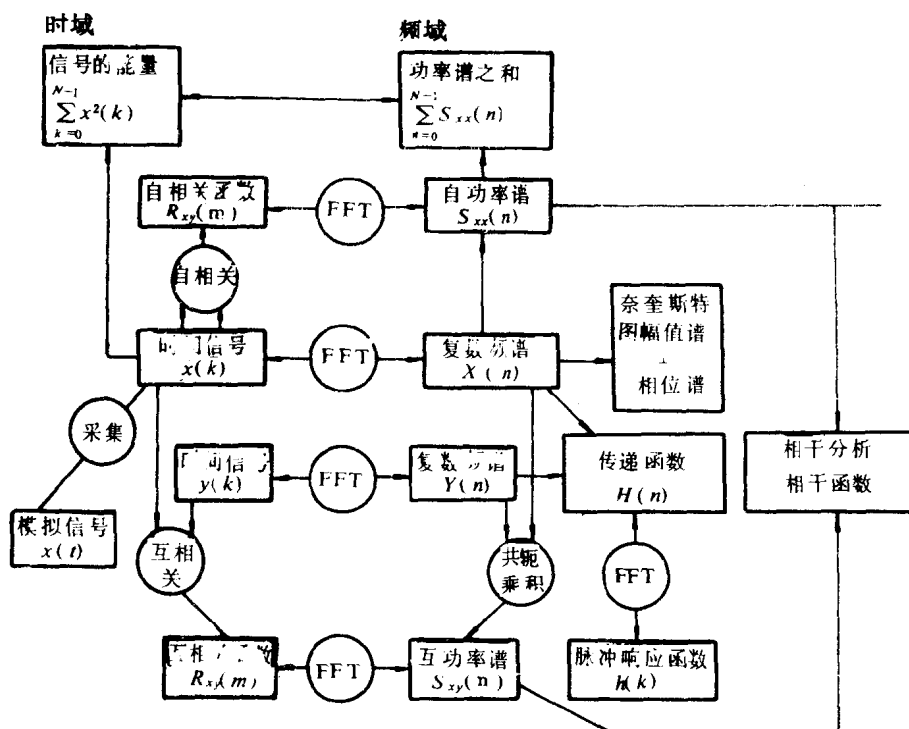


图 2-3 时域频域的相互关系

与离散互相关函数有关的各种关系可借助于图 2-3 来说明。 $x(k)$  与  $y(k)$  的互相关函数可以直接计算，如图中左侧所示。但这样计算很费时间，特别是当样本点数  $N$  较大时。另一种方法是采用 FFT 算法，首先算出两个信号的频谱，然后用一个频谱的共轭来乘另一个频谱，这一运算结果就是互功率谱，再用 FFT 对它作反变换就得到互相关函数。当两函数相同时，以上运算结果就是自相关函数。

FFT 的重要应用之四是用来计算功率谱。由图 2-3 可见，功率谱  $S(n)$  与相关函数  $R(m)$  是互为 FFT 对的关系。可以由已知的相关函数经过一次 FFT 变换得到功率谱；也可直接从原始数据  $x(k)$  作一次 FFT 求出频谱  $X(n)$ ，然后将其平方，最后乘以  $\frac{1}{N}$  获得自功率谱。