

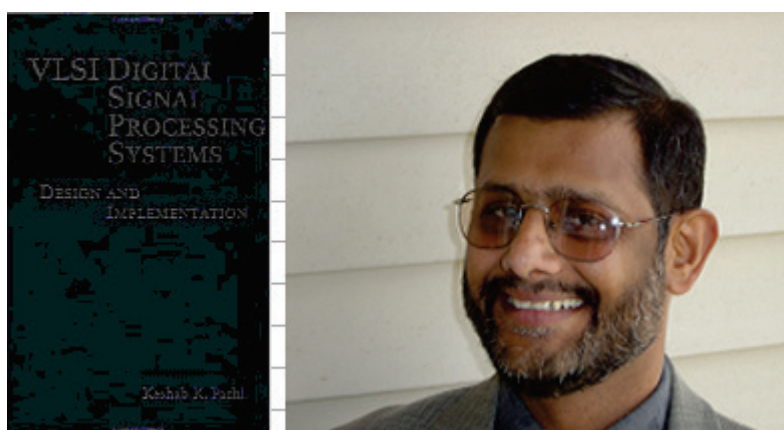
## 第一章、敲门砖——入门的准备

大家好，从这一贴开始，我们将陆续贴出 20 个帖子，作为帖子课堂，来和大家一起精读一本“经典”。也许有些同志表示怀疑：是什么书？值得这么搞吗？这里我先不发表自己的看法，直接援引两段精彩的书评来真实地反映了这本书的地位和内容。

1. "Globally there hardly exist more than a dozen book references on the subject of DSP hardware design. Among them...[Parhi's book is one of the] incontestable leaders, in both depth and breadth." (Analog Dialogue)
2. 视频压缩、无线通信、全球定位、雷达景像……在DSP的广泛应用领域内，怎样设计出高速、精巧的VLSI系统？并行处理、流水线、ASIC、可编程数字信号处理器，为实现DSP算法，技术与工具怎样结合才能更加完美？本书半为您提供答案。它既是折叠、异步流水线等技术的资源宝库，同时也是方向标，通过大量的实践经验指明了进行VLSI王国的捷径。

作为新手，也许你看懂了第一条，但对第二条不敏感。而高手应该非常赞同，甚至有如获至宝的感觉。

回想当初，国内还没引进这本书，我的导师就托自己的学生从海外火速寄回一本，看后疯狂推荐自己学生去看。第一眼看到这本书，我的感觉就是，很一般啊，甚至觉得封面有点丑，大家看看这图片就知道了。高手不要骂我哦，“人不可貌相，海水不可斗量”，怎么你到这岁数还不知道这点。骂得好，这不虚心地向大家推荐这本书呢！这里我把书的作者照片贴出来，以表示个人对他的崇拜之情。



书名：**VLSI Digital Signal Processing Systems: Design and Implementation**  
**VLSI 数字信号处理系统：设计与实现**

作者：**Keshab K. Parhi**

网址：<http://www.ece.umn.edu/~parhi/>

资源：<http://www.es.lth.se/ugradcourses/DSPDesign/>

幻灯片非常棒，帖子中的幻灯片 70-80%来源于这；

<http://www.cs.nctu.edu.tw/~ldvan/>

台湾一教授的课件，见 [Teaching](#) 目录下；

<http://homepages.cae.wisc.edu/%7Eece734/>

**压轴好戏**，国外一著名高校 vlsidsp 课程网站，学生项目很棒，相关的并行计算资

源很多。

直到现在，我们只知道作者和书名，书的内容是什么啊？大家别急，先来看看作者是何方神圣，了解一下他的水平，看值不值得去看他的书。

#### 作者简介

凯夏博·帕里(Keshab K. Parhi)教授分别于 1982 年、1984 年和 1988 年在印度理工学院、美国宾夕法尼亚大学和加州大学伯克利分校取得电机工程专业学士、硕士和博士学位。从 1988 年起，Parhi 教授在明尼苏达大学从事教学与研究工作，并被评为电机与计算机工程系的 McKinght 荣誉教授。

他的研究领域重点在于宽带通信系统物理层方面的 VLSI 体系结构设计。他当前的研究包括纠错码的编 / 解码器与加密 / 解密算法的体系结构、高速发送 / 接收器、超宽带系统、量子纠错编 / 解码器与量子 / 解密算法的体系结构。高速发送 / 接收器。超宽带系统 / 量子纠错编 / 解码器与量子加密 / 解密。他已经发表了 350 多篇学术论文，撰写了教科书“VLSI Digital Signal Processing Systems: Design and Implementation (Wiley, 1999)”。这套教材被美国、荷兰、瑞典、芬兰等国的多所大学采用。此外，他还参与了“多媒体系统的数字信号处理 (Marcel Dekker, 1999)”一书的编写工作。

Parhi 教授曾获得多次殊荣，包括 2003 年度“IEEE Kiyo Tomiyasu”技术领域奖。2001 年 IEEE W. R. G. Baker 最佳论文奖和 1999 年度 IEEE 电路与系统协会 Golden Jubilee 奖。1996 年，他被评选为电气电子工程师学会会士 (IEEE Fellow)。历年来，他曾担任“IEEE Transactions On Circuits and Systems, Circuits and Systems-II”、“VLSI Systems”、“Signal Processing”、“Signal Processing Letters”等学术杂志的编委会成员。此外，他还担任“IEEE Trans. On Circuits and Systems-I (2004-2005 年度)”的主编和“IEEE Signal Processing Magazine”的编委。Parhi 教授是 IEEE 1995 年“VLSI 信号处理研讨会”和 1996 年“专用系统、体系结构与处理器国际会议 (ASAP)”的技术委员会联合主席，以及 2002 年“IEEE 信号处理系统研讨会”的主席。他曾被 IEEE 电路与系统协会授予 1996-1998 年度杰出讲师的称号。

从这个作者简介中，我只记住两条：

- 1、世界级牛逼讲师；
- 2、电路系统牛逼强人。

(我再废话一段，就进入正题)

在我所拜读的书籍中，有一本堪称黄金宝典“深入理解计算机体系结构——程序员观点”，对软件工程师、嵌入式工程师甚至硬件工程师，肯定都对下面这句评论非常赞同：

**本书被誉为价值超过等重量黄金的无价资源宝库。**

那么在 DSP 硬件架构方面，Parhi 教授的这本《**VLSI 数字信号处理系统：设计与实现**》也配得上这句话。你想成为 VLSI-DSP 领域的顶尖高手吗？你想驾驭 VLSI-DSP 硬件架构的变换技术吗？读完这本书，掌握其中的技术，将使你力量敲开 VLSI-DSP 殿堂之门，荣登大雅之堂，此后你将拥有魔术般神奇的绝活，能变换出各种令人叹为观止的“DSP 系统”。

下面我们进入正题，也就是参考书的第一章“数字信号处理系统导论”，这一章分三个小节（我的划分方法，不一定完全与参考书相同）。

- 第一节、开胃菜（也就是 1.1 节 引言）。这里面有两个非常重要的常识点，故而单独列

为一节讨论。

- 第二节、一网打尽 DSP 算法。一网打尽，并不是说我会罗列所有 DSP 算法来做枯燥的讨论，甚至是书上的 10 个 DSP 算法我都不打算全拿出来讨论。之所以用一网打尽这个词，是因为想告诉大家，本书所介绍的技术确实覆盖了 DSP 几乎所有应用的各个方面。书中列举的 10 个应用示例，就是要让你知道，不论你做的是什么 DSP 应用，都难逃一死，都会被本书的技术直接摆平。所以，我们可以不去全看书上的这 10 个应用，挑跟你相关的看即可。在以后的章节中，都只会用那些最简单的 DSP 示例，比如 FIR、IIR 等等，把深奥的技巧轻松传授给各位。
- 第三节、主菜（DSP 算法的四种图形表示）。说白了，就是四种 DSP 表示图的物理意义和画法而已，但这却是本章的重点（但不是难点，本章没有难点）。大家务必把这四种图给掌握了，在以后的章节中，我们将用这些图来描述一个 DSP 程序，所有对 DSP 进行的变换（操作）都在这些图上进行。直到完成了 DSP 表示图的改造，才把这些表示图还原（映射为）硬件逻辑系统。之后就可以用 hdl 语言去描述和实现了。

说到这，我又想起实验室找工作的同学了。大家都喜欢做系统架构师，那工作钱多。可是你也得有本事才行，否则公司凭什么让你做系统架构师，你有何本事设计出高效经济的系统呢？本书将带你走上这条阳关大道。

讲解：第一节，开胃菜——引言

性能按以下指标来衡量：所需的硬件电路和资源（即占用的空间或面积）；执行的速度，这取决于吞吐率和时钟频率；功耗，或者说是完成某个给定任务需要的总能量。对于定点 DSP 系统，尤其是数字滤波器，有限字长的性能（即量化舍入噪声）是其第四个性能指标，因为一个舍入噪声大的数字滤波器即使在面积、速度以及功耗上有很好的性能也是无法使用的。

讨论：四个性能指标

#### 1、硬件资源消耗

这一点，实际工作的同志们应该很有体会。吝啬的老板总是让你用最少的资源做出复杂的系统（比如 H.264 编码）。大方的老板可能不这样要求，不过我们提倡大家勤俭节约，在满足应用要求的前提下，尽可能减少硬件资源的消耗。假设理论上，一块非常小规模 FPGA（也很廉价），恰好可以把一个 H.264 编码器做出来。你能做到吗？只要浪费一点点资源，就会发现资源不够用。恰好能做出来，这是一个极限，你的设计得非常精妙才行。不过大家不要灰心，本书的技术将有助于你做到这令人不可思议的一步，到时你就能灵活操控硬件资源消耗。

#### 2、执行速度

这就更容易理解了，为啥买机子都要上 2.0G 甚至 3.0G 的 CPU，不就是打游戏更爽一点吗？但是执行速度不仅仅由时钟速率（频率）来衡量。那么说吧，2.0G 双核与 3.0G 单核，你选哪一个？我们通常的看法，双核的相当于  $2.0G \times 2 = 4.0G$ ，而单核就只是 3.0G 而已，傻瓜都会选双核。由此看来，衡量系统性能只看一个时钟频率的确不明智，这里引入另一个指标，吞吐率。吞吐率指的是单位时间系统所能处理的数据量多少。再谈谈上面的例子，假设一个 CPU 核一个周期只能处理一个数据，那么 3.0G 单核 CPU 的吞吐率就是每秒处理 3.0G 个数据，而 2.0G 双核 CPU 吞吐率是每秒处理  $2.0G \times 2 = 4.0G$  个数据，显然双核吞吐率要比单核高。看来平时人们总喜欢用 时钟数  $\times 2$  来表征双核 CPU 的能力，原来就是想用吞吐率

来衡量 CPU 的执行速度（虽然不严格），但是从中可以发现，仅仅只用时钟频率来衡量系统执行速度是不全面的。对并行系统而言，电路时钟也许比串行系统低，但是一个时钟内处理数据的个数却比串行的多，此时用吞吐率（=时钟频率×并行度）也许更为合理。最后再明确一下，执行速度并不是单纯指时钟频率，有时可能指吞吐率。在以后的章节中，有时我们关注于时钟频率，有时又关注于吞吐率，或者两者都会关注，大家应该根据上下文去判断，去理解到底执行速度当时指什么。

### 3、功耗

功耗分静态功耗和动态功耗。很多同志都知道动态功耗和频率有关，频率降下来功耗自然就低了。但是除了频率的因素，功耗还与电路的电压以及电路电容有关，简化公式为

$$P = C \cdot V^2 \cdot f$$

实现同一功能的 DSP 系统，可以有不同结构，每种结构对应一定的电容量。假设频率  $f$  保持不变，电容越大，充放电时间越长，所需功能电压要更大才能保证一个周期内信号达到规定的低电平或高电平，相反电容小，那么供电电压就可以将下来，从而功率就降低了。再次提醒，这本书的技术使得你有能力去做各种变换来降低功耗。

### 4、计算精度

某些高级的 DSP 应用对计算精度要求很高。相信遇到这种情况的同志并不多，毕竟这是高级 DSP 应用了，但是如果掌握一套控制计算精度的方法，将有助于设计出更为出色的系统。

以上讲的是常考点一，系统的四个性能指标。大家可以根据实际工作情况对此进行补充。当然也欢迎你指正其中的错误。

### 讨论：两个重要特性

DSP 区别于其他通用计算的两个重要特性：实时吞吐率的要求和数据驱动的性质。

#### 1、实时吞吐率

实时吞吐率的需要和数据驱动的性质。实时吞吐率强调一个实时性，来一个数据得在规定的时间内处理完毕，不像某些计算先是存下来再处理。要做到实时，系统吞吐率“大于等于”数据采样率。注意“大于”就不需要了，一旦满足了采样率的要求，再加快速度也是没有意义的，因为没有那么多数据让系统处理，此时要考虑的不是让系统更快，而是让系统更省资源、更省功耗或者是计算精度提高才是。记住，只需吞吐率“等于”采样率，就可以达到实时要求，然后转而考虑其他性能方面的提高。

#### 2、数据驱动性质

在同步电路中，时钟负责系统运行的步伐（这就不说了）。而异步系统中却不一样，异步系统往往可以做得比同步系统更高效，但也更难设计。在异步系统中，系统各模块不必同时工作，而是由数据流来驱动，输入数据准备好了，就可以马上开始工作，数据处理完毕就可以停下来，以节省功耗。

注：书中第 16 章，对同步流水线、波流水线和异步流水线讲得非常有意思。这在我们以后的帖子中会讨论到。

-----第一节,结束,休息一下,休息一下-----

讲解：第二节，一网打尽 DSP 算法

这里我不打算详细讲解书中的各个 DSP 算法，那是数字信号处理课的内容，而我们要讨论的是 DSP 硬件架构的变换技术。之所以有那么一节，只想告诉大家，这里的一些应用，以后我们会做为示例来分析，所以你得先准备准备。要求不高，你对哪个应用感兴趣就去看。当然也欢迎你发帖聊聊你对某个应用的想法，或者是你正在做的 DSP 应用（不一定非得是课本中的应用）。可以说你的应用都用了那些 DSP 知识，都采用了什么技术解决了哪些难点问题（这一节主要就是闲聊，放松一下）。

*对我而言，还在学校念书，懂得也就是基本知识，所以请各位大侠能辅助我做好讲解，及时纠正鄙人错误之处，千万不要给我面子（题外话）。*

- 比如课本 1.2.1 的卷积和 1.2.2 的相关，这两个计算就是哥俩好，骨肉相连的，不说大家也知道，相关也就是特殊一点的卷积罢了。
- 1.2.3 的数字滤波器就更别说了，做 DSP 的能不知道 FIR 和 IIR 吗？
- 1.2.4 的自适应滤波器我没学过，但听说用途很广，很有用，属于随机信号处理的知识。啥时有空我得学学，或哪个同志发个帖子普及一下也可以。
- 1.2.5 的运动估计，做视频的同志再熟悉不过，从 MPEG-1/MPEG-2/MPEG-4 到 H.261/H.263/H.264，听说 08-10 年将推出 H.265，将是整合 MPEG-4 和 H.264 的优点，令人期待。运动估计是视频编码中最耗时的运算，如何用硬件加快其速度是非常热门的一个问题。
- 1.2.6 的离散余弦变换，这家伙也太多见了（如果你只知道 FFT，那么建议你在书上找找 DCT，把它看一遍），总爱抛头露面。不过近年来，出了个“小波”，开始和它抢饭碗了。
- 1.2.7 的矢量量化。做语音的同志常接触。我没做过语音，不过一看就懂（课本上的相应内容），建议大家也看看。
- 1.2.8 这个我不在行，以后用到再回来仔细研究，或者谁来补充一下也行。
- 1.2.9 的抽取和插值，还有 1.2.10 的小波和滤波器组，最近几年用的很火，很被看好，但我还没系统研究过，属于高级话题，用到再回来讨论。
- 格型结构（这是导师极力推荐看的）。的确，我也感觉到了，新的 DSP 教科书越来越重视格型结构，将其列为单独一张讲解，例如清华的胡广书老师的《数字信号处理》就讲的很透彻，感兴趣的同志可以去看看。

第二节我们就聊那么多，大家可以发贴补充。以后用到，我会详细的讲解，现在讲多了，以后也记不得（^\_^）。

-----第二节,结束,休息一下,休息一下-----

讲解：第三节，主菜（DSP 算法的四种图形表示）

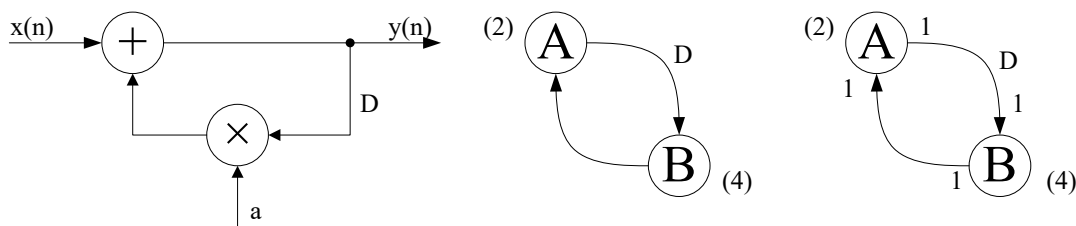
第三节内容不多，就是教你如何画出四种 DSP 表示图，但地位及其重要。以后的章节就是采用这些表示图来进行 DSP 的描述和变换（操作）的。如果你现在看得不是很懂，请你以后用到之前务必回来这再仔细看看，一定要在用之前完全看懂，否则非常不利于学习掌握。



先看第一个：**框图**。框图和硬件线路图非常接近，也比较直观，数字信号处理教科书上最常用的就是框图。因为大家都熟悉，所以我也不多说，注意的是， $Z^{-1}$ ，表示一个延时。

往下，是**信号流图（SFG）**。看起来 SFG 就是框图的简化版本（第一个想出该图的可能是个懒鬼？），但是大家也许不知道，SFG 对于某些系统架构的变换非常方便，而其他形式的图却做不到。比如书中所说的转置，就是非常有用的系统架构变换之一，变换前后系统的功能不发生变化，但是新结构也许会带来更多好处，除了转置，还有类似的很多系统架构变换都需要用 SFG 来表示和操作（这个作为练习，大家发帖子说说还有那些变换定理）。反正，大家记住一点，对于某些系统架构的变换（如转置），用 SFG 表示和操作就非常方便，而其他表示图则不然。当然了，SFG 和框图是可以互相转化的，它们都比较清楚的表达出每个信号和功能单元。

在下来就是重点中的重点了，**数据流图（DFG）**。前面两种形式的图在数字信号处理或者电路理论中都时常见到，而 DFG 却很陌生，但在本书中却用得最为频繁。DFG 比前面讲的框图和 SFG 来得简单，如下图示，a)为框图，b)和 c)为 DFG。



a) 运算  $y(n)=ay(n-1)+x(n)$  的框图表示

b) 常规的DFG表示

c) 同步DFG表示

在 b)图中看不到输入  $x(n)$  和输出  $y(n)$ ，乘数  $a$  也看不见了。大家也许觉得奇怪，省去那么多东西，这还能全面表示 DSP 系统吗？其实，DFG 本来就不是用来“全面”描述 DSP 系统的，仅仅给你一个 DFG，也是不能还原出硬件框图。比如 b)中的 A/B 两个节点，直接看根本不知道对应什么计算，乘法器的系数  $a$  也丢失了，输入输出在哪边也无从知晓，所以说 DFG 不是用来“全面”描述 DSP 的，它是专用于突出数据之间的依赖关系和节点计算时间的。

仔细阅读课本 1.4.3 节的描述就知道怎么来画 DFG。首先，框图中的一个计算节点在 DFG 中也对应一个节点（名字有你来取），如 b)中的 A 节点对应 a)中的加法节点，而 b)中的 B 节点对应 a)中的乘法节点。然后，从 A 到 B 有一个数据依赖关系，就是 A 的输出经过一个延时作为 B 的输入，用从 A 指向 B 的有向边表示，延时直接作为边的权值标注出来（比如可用 1、2、3 或者是 D、2D、3D 来分别表示 1、2、3 个延时）；另外，B 到 A 也存在一个依赖，即 B 的输出直接作为 A 的输入，从 B 指向 A 的边延时为 0。最后把 A/B 节点的计算时间用小括号括住写在节点旁边。

DFG 很容易画，但是其物理意义我们要清楚，这里涉及到两个术语：“迭代内优先约束”和“迭代间优先约束”。

- 迭代间优先约束关系：比如图 b)，从 B 到 A 的有向边，延时为 0，这表示 B 优先于 A。B 计算完成输出结果传到 A，A 才开始“有意义”的计算并得出结果（之所以加上“有意义”限定词，表明 B 没计算完成之前，A 要是有什么计算的动作都是没有意义的，仔细思考一下实际硬件的物理过程就明白了）。这种 B 优先于 A（或说 A 依赖于 B）的关系称为迭代内优先关系。迭代内，指的就是一个周期内部，很多同志对“迭代内”三个字不感冒，那么请你暂时记下，因为这三个字很关键，是必

须要理解的。

注：熟悉 verilog 的同志知道这种迭代内优先约束关系在 verilog 中对应什么吗？这就是组合逻辑的阻塞赋值语句啊。前面一条语句描述 B 的计算，后面一条语句描述 A 的计算。也就是说 B 的计算要先执行，然后才到 A 的计算。

- 迭代间优先约束关系：不同周期期间的优先关系，比如图 b)，从 A 到 B 的一条带一个延时的边，表示 A 迭代间优先于 B，注意这里不同于迭代内的优先关系，这里指的是 B 的第 n 次迭代要等 A 的第 n-1 次迭代执行完毕方可开始，因为 B 的第 n 次迭代输入所需就是 A 第 n-1 次迭代的输出，这是不同节点的不同迭代间存在依赖关系。容易理解，A 的输出确实要经过一个延时才到达 B。

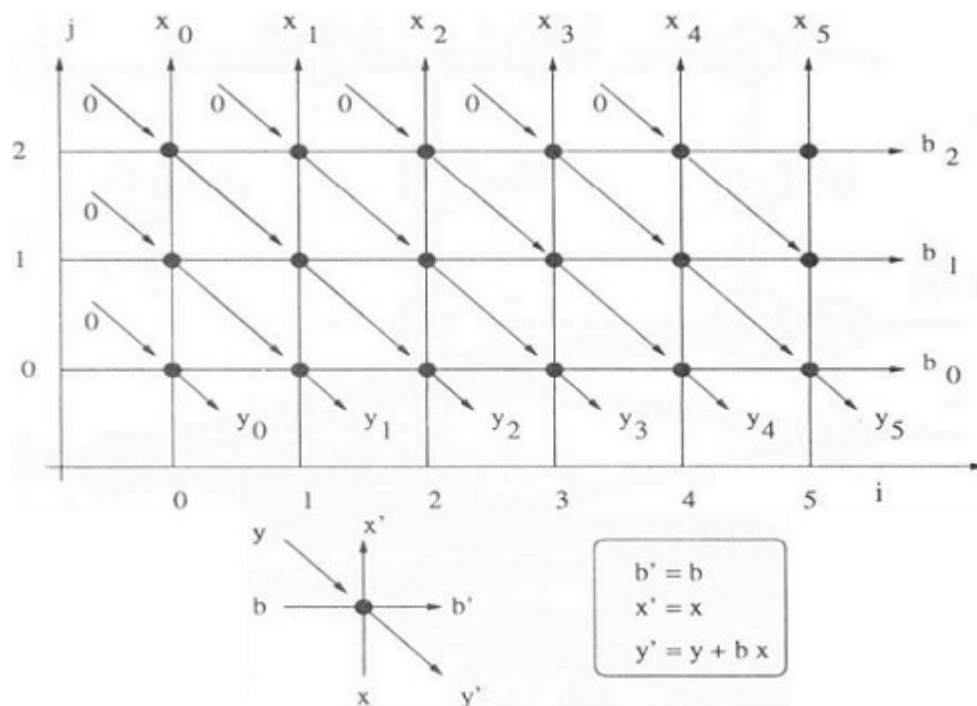
课本中有一段话比较重要，大家可以仔细品味品味：DFG 主要用在高层次综合中，它将 DSP 应用的并发实现方法推导到并行硬件上，其中子任务的调度以及资源分配是主要关心的目标（调度是指确定节点何时并且在哪个硬件单元中执行）。

基于我目前的理解，DSP 本质是一个调度问题。在保证系统功能不变的情况下，增加或去掉某些节点之后应该如何进行调度，或者是如何进行调度，可以使得关键路径最小等等。用调度的观点去看待 DSP，对以后的学习很有帮助。

接下来，课文中还对单速率的 DFG 做了扩展，提到了同步数据流图（SDFG）、单采样率 DFG（SRDFG）和多采样率 DFG（MRDFG）。MRDFG 就是单节点 DFG 的扩展，这里我只提示一个词“最小公倍数”，大家就作为练习看看课本对图 1-25 的描述，想想那边用到了最小公倍数，这个就算明白了。

最后一个要介绍的是**依赖图（DG）**。DG 与 DFG 类似，都是描述节点间的优先约束关系。在 DFG 中，我们详细讲解了“迭代内优先约束关系”和“迭代间优先约束关系”，DG 中也有这些关系，但是它们俩的不同在于：DFG 中的节点只涵盖了对应算法的一次（仅仅一次）迭代中的计算，而 DG 涵盖了所有次迭代的运算，故而课本中说到，DG 中，算法每调用一个新的运算，就会产生一个新的节点。DSP 程序会不停的调用一个物理运算节点来计算，这样这个物理运算节点在 DG 中将对应多个节点，但是 DG 中对应同一个物理运算节点的节点存活在不同的周期内（这么说能理解吗？有时候真想与大家面对面的交流，一下就懂了，用文字表述有时词不达意，故请大家多多包涵，自己多看看书）。

接下来，看看 DG 和 DFG 的区别。DFG 有延时单元，DG 也有，但 DG 的是隐藏起来的，不像 DFG 直接标注出来，DG 使用边所跨越的时间长度来表示延时的个数（看完就知道了）。



三阶FIR迭代方程  $y(n)=b_0*x(n)+b_1*x(n-1)+b_2*x(n-2)$  的依赖图

大家注意看上图的 DG 和其对应的迭代方程。水平轴  $i$  表示时间，标明了第  $xxx$  周期，图中画出第 0 周期到第 5 周期，后面就省略了。纵轴  $j$  表示计算节点，图中就只包含 3 个物理计算节点，记为节点 0、节点 1 和节点 2。可是明明图上有很多节点啊，怎么说只对应于 3 个物理计算节点？DG 是描述多次迭代的，不同于 DFG。所以在时间方向，也就是横向一行节点都对应同一个物理计算节点，图中不就只有三行节点吗？就分别对应于节点 0（最低行）、节点 1（中间行）和节点 2（最高行），同一行节点对应于同一个物理计算节点的不同周期的调用。每个节点都表示一次乘加运算，将水平向右的数与竖直向上的数相乘，在与第二象限斜向下的数相加，得到第四象限斜向下的数。比如最低行所有节点都使用乘数  $b_0$ ，所以有一条水平向右（方向  $(1, 0)$ ）的线贯穿这行节点。中间行和最高行节点类似，它们两的乘数分别是  $b_1$  和  $b_2$ 。纵向向上（方向  $(0, 1)$ ）的线，对应  $x_0, x_1, x_2, \dots, x_5$ ，表示不同时刻的输入，对于周期  $i$ ， $x_i$  同时穿过一系列节点，也就是说周期  $i$ ， $x_i$  会同时传递到节点 0，节点 1，和节点 2，作为它们的输入进行运算。斜对角线向下（方向  $(1, -1)$ ）的线，比如穿过节点  $(1, 2)$  的斜线，表示在周期 1，将  $0$  与  $x_1*b_2$  相加，然后延时一个周期（到周期 2）传到节点  $(2, 1)$ ，也就得到  $x_1*b_2+x_2*b_1$ ，同理再延时一个周期（到周期 3）传到节点  $(3, 0)$ ，就得到  $x_1*b_2+x_2*b_1+x_3*b_0$ ，这个也就是输出  $y_3$  了。

DG 是描述分析多次迭代中优先约束关系的最佳工具，主要用于脉动阵列的设计。脉动阵列是一种 MISD 结构，如果把本书的技术当成兵器的话，我称脉动阵列为手雷，可以杀伤一片。脉动阵列的实现就是规则的一片。

注：虽然讲完这一节，我感觉还是讲的不够清楚。授人予鱼，不如授人予渔，这里我把自己看书的方式告诉大家，你们也可以试试看行不行得通。比如说 DFG 和 DG 这两个的内容，这两者都有示例，DFG 的是  $y(n)=a*y(n-1)+x(n)$ ，DG 的是  $y(n)=b_0*x(n-2)+b_1*x(n-1)+b_0*x(n)$ 。我是先把这两个式子的实际迭代计算的过程弄清楚了，然后再看书，再去看看 DFG 和 DG 是怎么来描述实际的迭代过程的，以及他们描述了迭代过程的什么方面。对于这一节讲到的四种表示图，它们各有各存在的理由，都有其用武之地，所以请大家务必理解。



-----第三节,结束,休息一下,休息一下-----

本章任务完成,我们做个小结。首先是开胃菜一节,讨论了 DSP 的四个重要性能指标:资源、速度、功耗和精度。此外还讨论了两个 DSP 的独有特性,实时吞吐率和数据驱动性质。第二节,我们闲聊了一些 DSP 的典型调用,如果你明白了只要是 DSP 应用,本书技术就能应用到其中,目的也就达到了。第三节是重点,介绍了 DSP 的四种表示图:框图、SFG、DFG 和 DG。随着学习的深入,我们将逐步吃透它们,明白它们的特点及应用场合。我保证你会喜欢他们的,它们会让你的学习过程充满乐趣。

说了那么多,我一直都没给大家详细介绍这本书的内容,为什么呢?因为我想让你在看帖子的时候去思考,到底这是怎么样的一本书,书中蕴含了什么样的宝藏。所以虽然没直说书的内容,但字里行间却向大家传递书中有宝藏的信息,以激发大家的学习兴趣。这里我再援引前面出现过的一段评论,让你领略本书的冰山一角:

视频压缩、无线通信、全球定位、雷达景像……在DSP的广泛应用领域内,怎样设计出高速、精巧的VLSI系统?并行处理、流水线、ASIC、可编程数字信号处理器,为实现DSP算法,技术与工具怎样结合才能更加完美?本书半为您提供答案。它既是折叠、异步流水线等技术的资源宝库,同时也是方向标,通过大量的实践经验指明了进行VLSI王国的捷径。

怎么样,看明白了吧!通俗地说,这本书通篇“折衷之术”,“折衷”是一门高深的哲学(google见《文心雕龙》)。它教会你,如何在DSP系统的各个指标上进行灵活的折衷,教会你如何玩弄DSP的系统架构。