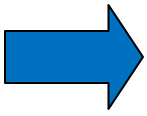
硬件设计新手入门宝典

之第二部

微信公众号‘超硬工程师’，定期以连载方式系统地分享发布嵌入式，计算机系统硬件电路及系统设计知识和调试经验。

微信上扫描下面二维码或搜索公众号‘超硬工程师’，关注后就可学习，还可免费申领《硬件设计新手入门》系列丛书。

学硬件设计当然选择  **微信公众号‘超硬工程师’**

学硬件设计，关注微信号：超硬工程师

目　　录

前 言

大家都知道，硬件设计行业除了一些教授电路基本原理的书籍之外，没有专门的讲解专业硬件及系统产品开发技术及流程的书籍。对于从事相关行业的朋友，提高更多是通过前人的传授和个人产品研发经验的累积。为给从事硬件设计行业的朋友提供快捷高效的提升个人能力的渠道，微信公众号‘超硬工程师’全面系统地汇总硬件及系统设计知识和经验，并免费分享，从而帮助更多的从业者快速掌握获取相关知识和经验，在硬件设计职业生涯之路上达到全新的高度。

‘超硬工程师’微信号平台建立后，不少网友留言对未来觉得迷茫--学生朋友说学校里学不到实用知识，快毕业了还感觉什么都不会；参加工作的年轻朋友觉得在单位一直打下手，又没人指点，不知怎么入门。大家都非常期望能够有人带一带。超硬工程师觉得应当尽绵薄之力，帮助大家学有所成，尽快建立一些基本的设计能力。因此通过连载的形式，发布一系列文章，详细剖解硬件设计基础知识和设计过程。

之前发布过精读 datasheet 系列文章，相信大家看了之后，应该知道如何去阅读元器件的技术文档了。做硬件设计就是把不同元器件连接在一起，结合一些软件以实现产品设计所要达到的功能。所以硬件设计另外一个重要的能力就是去读懂硬件

设计电路原理图为将来设计硬件电路打下基础。这个系列的文章就以一份基于 X86 平台的嵌入式设计为基础，来详细解析一下硬件电路原理图设计，常用电路及基本功能原理。大家跟着这个系列走下去，相信会很快提高你阅读理解硬件电路的能力。欢迎大家在微信上搜索，或扫下面二维码关注微信号：超硬工程师。

第四章　CPU 芯片及功能解析

我们都知道处理器 CPU 是一个计算机系统的核心，其在计算机系统里的作用就想人的大脑对于人体的作用一样关键。这一章就来介绍一下前一章中讲到的系统框图设计中用到的 Intel Atom 凌动处理器（N450 或 D510），让大家对 CPU 芯片功能有个更加深刻的认识和理解。

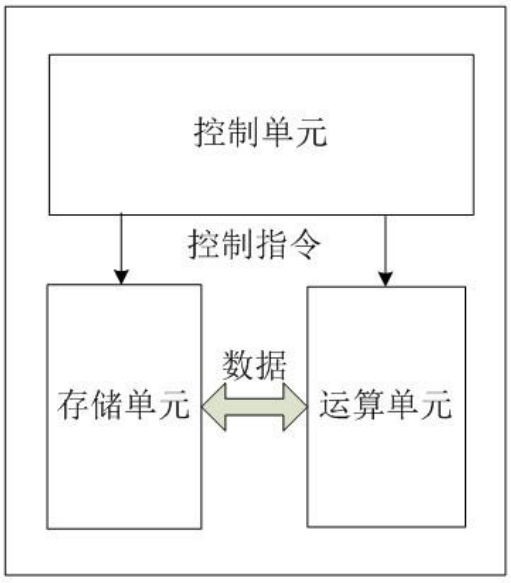
一， CPU 架构及工作原理

本文我们以 Intel 为例对 CPU 的工作原理做简单介绍。

1971 年，Intel 公司的工程师特德·霍夫发明了世界上第一个微处理器—4004，这款 4 位微处理器虽然只有 45 条指令，而且每秒只能执行 5 万条指令。Intel 从 8086 开始，就进入了我们所谓的 x86 时代。而 80386 的诞生则标志着 Intel 正是进入了 32 位微处理器的时代。从 80386 到 Pentium 4 这个年代的 CPU，就是传说中的 IA-32 时代。Intel 公司的 CPU 发展历程如下表所示：

型号	位宽	总线位宽	地址位	寻址空间	备注
4004	4	4		640B	1971-11-15
8008	8	8		16K	1972-4-1
8080	8	8	16	64K	1974-4-1
8086	16	16	20	1M	1978-6-8 X86 起源
8088	除了外部总线位宽是 8 位，其他参数和 8086 完全一致				
80186	16	16	20	1M	简单认为较 8086 多了几条指令而已
80286	16	16	24	16M	多任务、多用户系统的核心
80386	32	32	32	4G	包含了分页的虚拟内存机制
80486	32	32	32	4G	
Pentium 1,2,3,4,酷睿，酷睿 2					

我们都知道 CPU 的根本任务就是执行指令，对计算机来说最终都是一串由“0”和“1”组成的序列。CPU 从逻辑上可以划分成 3 个模块，分别是控制单元、运算单元和存储单元，这三部分由 CPU 内部总线连接起来。如下所示：



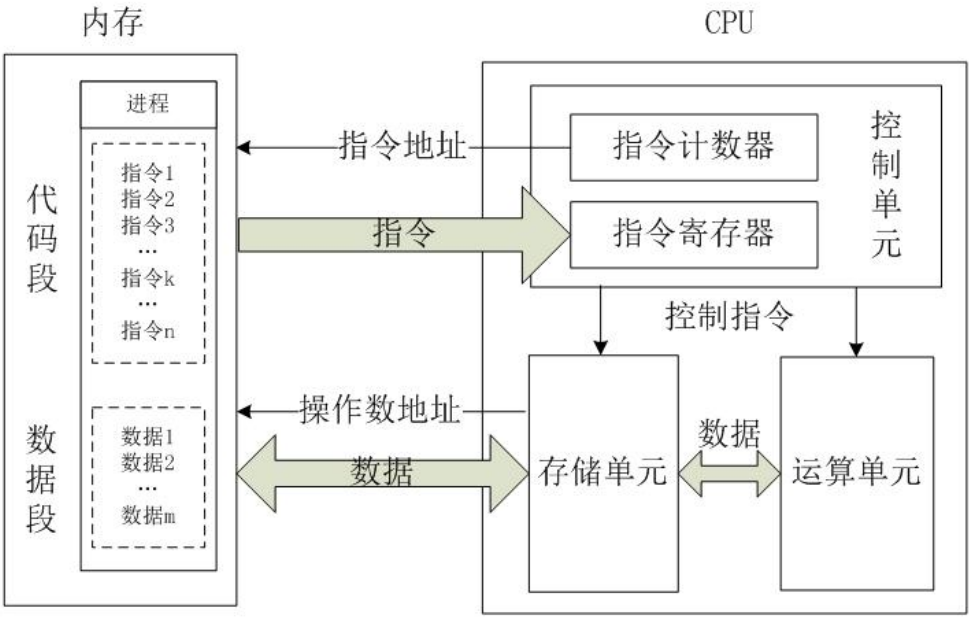
控制单元：控制单元是整个 CPU 的指挥控制中心，由指令寄存器 IR(Instruction Register)、指令译码器 ID(Instruction Decoder)和操作控制器 OC(Operation Controller)等，对协调整

个电脑有序工作极为重要。它根据用户预先编好的程序，依次从存储器中取出各条指令，放在指令寄存器 IR 中，通过指令译码(分析)确定应该进行什么操作，然后通过操作控制器 OC，按确定的时序，向相应的部件发出微操作控制信号。操作控制器 OC 中主要包括节拍脉冲发生器、控制矩阵、时钟脉冲发生器、复位电路和启停电路等控制逻辑。

运算单元：是运算器的核心。可以执行算术运算(包括加减乘数等基本运算及其附加运算)和逻辑运算(包括移位、逻辑测试或两个值比较)。相对控制单元而言，运算器接受控制单元的命令而进行动作，即运算单元所进行的全部操作都是由控制单元发出的控制信号来指挥的，所以它是执行部件。

存储单元：包括 CPU 片内缓存和寄存器组，是 CPU 中暂时存放数据的地方，里面保存着那些等待处理的数据，或已经处理过的数据，CPU 访问寄存器所用的时间要比访问内存的时间短。采用寄存器，可以减少 CPU 访问内存的次数，从而提高了 CPU 的工作速度。但因为受到芯片面积和集成度所限，寄存器组的容量不可能很大。寄存器组可分为专用寄存器和通用寄存器。专用寄存器的作用是固定的，分别寄存相应的数据。而通用寄存器用途广泛并可由程序员规定其用途，通用寄存器的数目因微处理器而异。这个是我们以后要介绍这个重点，这里先提一下。

我们将上图细化一下，可以得出 CPU 的工作原理概括如下：



总的来说，CPU 从内存中一条一条地取出指令和相应的数据，按指令操作码的规定，对数据进行运算处理，直到程序执行完毕为止。

早期 Intel 的微处理器，诸如 8085，8086/8088CPU，普遍采用了地址总线和数据总线复用技术，即将部分(或全部)地址总线与数据总线共用 CPU 的一些引脚。例如 8086 外部地址总线有 20 根，数据总线复用了地址总线的前 16 根引脚。复用的数据总线和地址总线虽然可以少 CPU 的引脚数，但却引入了控制逻辑及操作序列上的复杂性。所以，自 80286 开始，Intel 的 CPU 才采用分开的地址总线和数据总线。

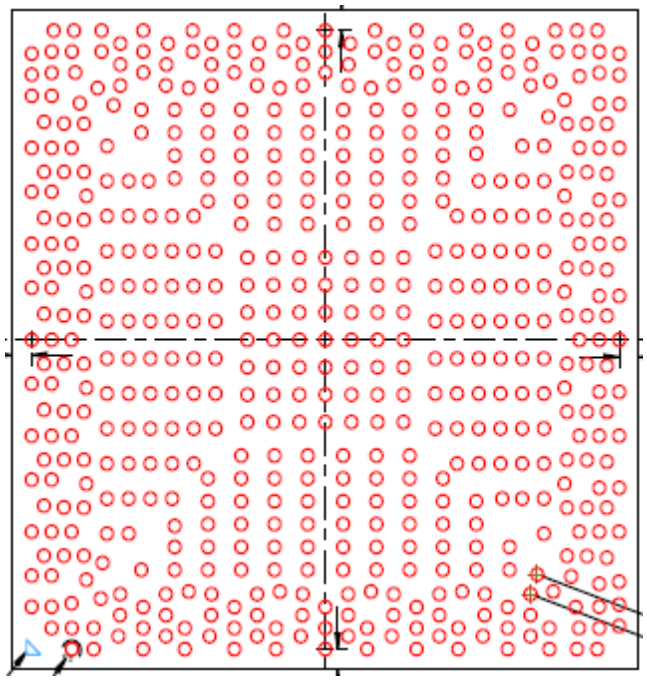
不管是复用还是分开，对我们理解 CPU 的运行原理没啥影响。总结一下，CPU 的运行原理就是：控制单元在时序脉冲的作用下，将指令计数器里所指向的指令地址(这个地址是在内存里的)送到地址总线上去，然后 CPU 将这个地址里的指令读到指令寄存器进行译码。对于执行指令过程中所需要用到的数据，会将数据地址也送到地址总线，然后 CPU 把数据读到

CPU 的内部存储单元(就是内部寄存器)暂存起来，最后命令运算单元对数据进行处理加工。周而复始，一直这样执行下去。

二， N450/D510 CPU 的主要特征及接口

熟悉英特尔 x86 系统架构的朋友都知道，传统英特尔硬件系统平台包含 3 个主要芯片：处理器，北桥（GMCH – 显示/内存控制器），以及南桥（ICH – 输入输出控制器）。基于 N50/D510 CPU 的硬件平台则不同，该硬件平台包含两个主要芯片：处理器和输入输出控制芯片组。这样的基于两个芯片设计的硬件平台可以提高性能，降低成本，简化设计/测试流程。

该处理器系列内部包含集成内存控制器（IMC），集成图形处理控制器（GPU），以及集成输入/输出接口控制器（IIC）。这些功能都都集成在一个硅片单元的裸片（die）上。该芯片的封装是 BGA559，采用 45 纳米的生产制程工艺。下图就是该芯片底部的引脚分布示意图。



下面列出该处理器芯片的主要特性：

- 单个 die (所有功能集成在一个硅片单元裸片)
- 32KB 指令缓存，24KB 回写数据缓存
- 支持 2 个线程的多线程技术（hyper threading）
- 512KB，8 路的二级缓存（L2 cache）
- L2 cache 二级缓存容量大小动态调控
- 支持 32 位及 64 位架构
- 双温度监控传感器 Thermal Monitor #1 & #2

该处理器芯片的主要总线接口：

- 系统内存接口
    - 支持 DDR2 SDRAM 内存（N450 只支持 DDR2）
        - 单通道（single channel），64 位
        - 最多支持 2 根 SO-DIMM
        - 内存接口的工作频率是 667MHz，数据传输带宽是 5.3GB/s (PC-5300)
        - 内存接口的工作电压为 1.8V
        - 只支持 non-ECC, un-buffered 的内存
        - 支持 512Mb, 1Gb, 2Gb 的内存颗粒
        - 最大支持 2GB 的内存容量
        - 支持内存温度管理技术。当芯片内的温度传感器达到一定温度时，内存温度管理机制启动，处理器对内存选择性地进行内存读写操作
    - 也可支持 DDR3 SDRAM 内存
        - 工作频率也是 667MHz
        - 也是单通道（single channel），64 位
        - 内存接口的工作电压为 1.5V
        - 最大支持 2GB 的内存容量

        - 最大两根 SO-DIMM

- DMI (Direct Media Interface) 接口
    - CPU 通过该接口及相关信号与芯片组 ICH 点对点相连
    - 单向 4 个 DMI lanes (一个 lane 就是一个差分信号对)，双向总共 4 个 DMI lanes
    - 每个 lane 的数据传输带宽是 2.5Gb/s，考虑到 8b/10b 转换，实际带宽是 250MB/s
    - 所以，单向 4 lanes 的带宽是 1GB/s；双向 8 lanes 的总带宽是 2GB/s
    - 接口的参考时钟是 100MHz
    - 支持 APIC 消息传输
    - 支持消息中断（Message Signaled Interrupt）数据传输
    - 支持电源管理状态信息数据传输
    - 支持系统管理管理中断(SMI)，系统控制中断(SCI)，系统错误(SERR)信息数据传输
    - 支持传统 ISA 协议设备的数据传输，比如并口，软驱，LPC 总线设备等

- 集成显示控制接口
    - 支持英特尔动态显存管理技术
    - 兼容 Directx 9
    - 使用 200MHz 时钟信号
    - 支持两种显示端口：LVDS 和 RGB
        - 数字端口 LVDS 的分辨率可达 1280*800 或 1366*768

        - 模拟端口 RGB 的分辨率可达 1400*1050，60Hz 的刷新频率
    - 支持 MPEG2 硬件加速

该处理器芯片的电源管理：

- 支持 ACPI 电源管理标准中定义的处理器的各种状态：C0, C1, C2, C4
- 支持 ACPI 电源管理标准中定义的系统的各种状态：S0, S3, S4, S5
- 支持系统温度管理(thermal management)模式

三， N450/D510 CPU 的主要功能

1，CPU 集成的内存控制

该处理器（N450 或 D510）内部的集成内存控制器可支持 DDR2 或者 DDR3 技术的内存。当然在板极设计中只能选择使用其中一种，无法在同一块电路板上同时设计两种内存技术。也就是说要主板设计成 DDR2，就不能支持 DDR3；设计成 DDR3，就不能支持 DDR2。

该内存控制器支持 64 位宽的，单个内存通道（single channel）。这个内存通道上最多可放置两根 non-buffer，non-ECC 的 SODIMM 内存插槽。

1） DDR2 内存设计

- 前面接口部分已经提到过该集成内存控制器的 DDR2 接口及总线的运行频率是 667MHz, 带宽为 5.3GB/s，也就是常说的 PC-5300 内存

- 内存颗粒：该内存控制器可支持容量为 512Mb/1Gb/2Gb 的，x16 的（数据位数为 16 位）内存芯片。
- 内存条的要求 – 支持两种类型
  - A 型：2 个 rank 的 x16（16 位数据线）SDRAM（双面）
  - C 型：1 个 rank 的 x16（16 位数据线）SDRAM（单面）
  - 不支持内存条的正反两面内存容量大小不一致。也就是说一面放置一定容量的内存颗粒，另外一面要么为空，要么放置同等容量的内存颗粒
  - 该控制器支持两根 SO-DIMM，考虑内存总线信号完整性，建议两个 SO-DIMM 使用相同类型

下图列出了所支持的 SO-DIMM DDR2 内存条配置。

Raw Card Type	DIMM Capacity	DRAM Device Tech.	DRAM Organization	# of DRAM Devices	# of Ranks	Page Size
A	512MB	512Mb	32M x 16	8	2	8K
A	1GB	1Gb	64M x16	8	2	8K
C	256MB	512Mb	32M x 16	4	1	8K
C	512MB	1Gb	64M x 16	4	1	8K
C	1GB	2Gb	128M x 16	4	1	8K

2） DDR3 内存设计

- 该集成内存控制器的 DDR3 接口及总线的运行频率是 667MHz, 带宽为 5.3GB/s，也就是常说的 PC-5300 内存
- 内存颗粒：该内存控制器可支持容量为 1Gb/2Gb 的，x16（数据位数为 16 位）或 x8（数据位数为 8 位）内存芯片。
- 内存条的要求 – 支持两种类型

- A 型：2 个 rank 的 x16（16 位数据线）SDRAM（双面）
- B 型：1 个 rank 的 x8（8 位数据线）SDRAM（双面）
- 不支持内存条的正反两面内存容量大小不一致。也就是说一面放置一定容量的内存颗粒，另外一面要么为空，要么放置同等容量的内存颗粒
- 该控制器支持两根 SO-DIMM，考虑内存总线信号完整性，建议两个 SO-DIMM 使用相同类型

下图列出了所支持的 SO-DIMM DDR3 内存条配置。

Raw Card Type	DIMM Capacity	DRAM Device Tech.	DRAM Organization	# of DRAM Devices	# of Ranks	Page Size
A	1GB	1Gb	64M x 16	8	2	8K
A	2GB	2Gb	128M x16	8	2	8K
B	1GB	1Gb	128M x 8	8	1	8K
B	2GB	2Gb	256M x 8	8	1	8K

2，N450/D510 处理器的集成显示控制器

图形处理是一个非常复杂的课题，这里只是简单介绍一下该处理器中集成的图形处理单元(GPU - Graphic Process Unit)的主要结构和功能。

该处理器集成的图形处理控制器包含了几个主要组件：图形处理引擎，像素处理，显示管道，显示端口。图形处理引擎通过内存控制器输入需要处理的图形数据，进行相关图形处理后输出图形所对应的像素数据到显存缓冲区，交由处理器进行数据编排处理。

- 图形处理引擎
  - 3D 引擎（3D engine） - 该集成显示控制器的 3D 引擎使用很长的流水线架构，流水线的每个步骤可以同时运行，从而可以极大地提高性能。3D 引擎主要负责对 3D 图形数据进行处理，包括顶点定位处理，光栅化处理，纹理帖图，像素处理等，最终完成 3D 图形的生成，并将图形映射到相应的像素点上。
  - 视频处理引擎（video engine） - video engine 主要处理非 3D 应用场景的图像数据，主要包括 VLD 和 MPEG2 的硬件解码。该引擎支持硬件运动补偿（HMC - hardware motion compensation）。HMC 是指通过参考图形来预测一个图形的像素和颜色从而重建新图形。GPU 收到视频数据流后通过 video engine 硬件进行图形运动补偿处理，这可以降低 MPEG2 软解码的工作，从而提高系统性能。
  - 2D 引擎（2D engine） - 2D engine 主要处理 2D 的图形数据，它通过一系列寄存器来定义 VGA 标准及其它标准中规定的不同图形模式所要求的图形数据特性比如颜色深度，分辨率，硬件加速等等。2D engine 支持 128 位的像素数据的块传输，从而提高 Windows 操作系统的图形接口的性能。

- 显示端口

该图形控制器包含两个显示管道：A 和 B。每个显示管道包含一组像素数据和时序发生器。时序发生器给每个显示管道提供图象显示所需要的时序信息。由于该图形控制器有两个显

示通道，因而可以支持两个独立图形数据流。图形数据流通过显示管道最终到达显示端口。

显示通道的终点就是显示端口，图形像素数据就是通过显示端口最后成像到外围显示设备上。这里的图形控制器支持两个显示端口：一个模拟端口 VGA，一个数字端口 LVDS。

- 模拟端口 VGA
  模拟显示端口输出红，绿，蓝（R,G,B）模拟信号以及行同步信号（HSYNC）和场同步信号(VSYNC)。除此外，还有 DDC 信号对（DDC_CLK 和 DDC_DATA）用于图形控制器和外围显示器之间配置和控制通讯，包括即插即用的设备识别和配置。

- 数字端口 LVDS
  该图形控制器也支持一个数字显示端口 LVDS(low voltage differential signal)，这个端口有一个 LVDS 传输通道，包含 3 根数据信号对，和一个时钟信号对。这个 LVDS 传输通道运行的时钟频率为 25MHz~112MHz。

该图形控制器支持两个显示端口，因而可以支持在两个不同的显示设备上显示不同图象内容。N450/D510 处理器中集成图形处理控制器支持两种显示模式：复制显示模式（两个外接显示设备显示同样图形内容）；扩展显示模式（两个外接显示设备显示不同图形内容，合在一起组成一个大的显示设备）。

3，N450/D510 处理器的 Hyper-threading 技术

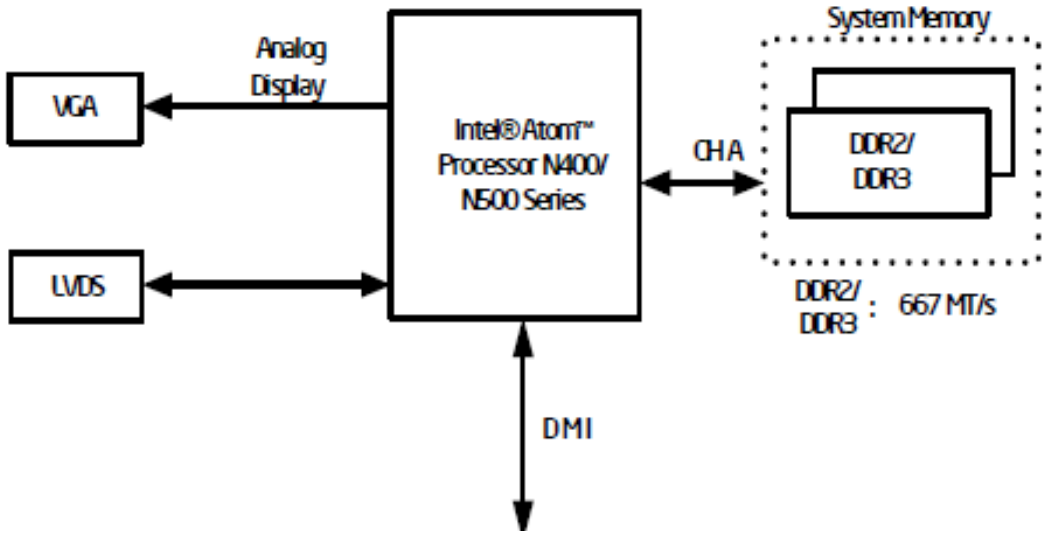
N450/D510 处理器支持 Hyper-threading 技术，允许一个物理处理器分成两个逻辑处理器。两个逻辑处理器共享一些资源，比如缓存，执行单元，总线等，但也有自己独立的一些资源比如通用寄存器，控制寄存器等。Hyper-threading 功能必须要在 BIOS 的设置中打开，并需要操作系统支持。

CPU 芯片的结构及功能就初步介绍到这里。下一章将会对 CPU 芯片引脚的信号连接及相关电路设计进行详细讲解。

第五章　CPU 芯片信号连接及电路设计

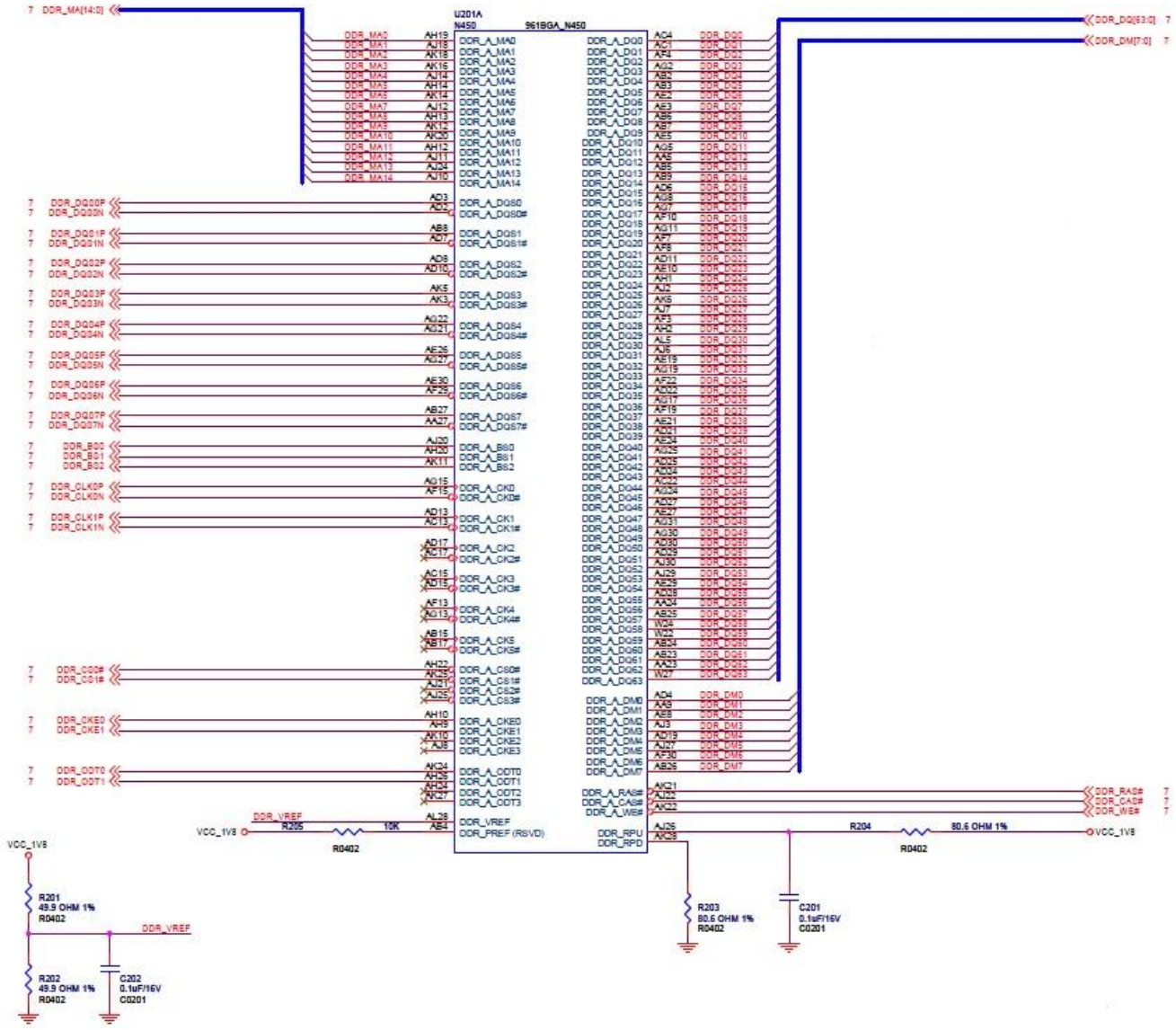
前一章讲了一下 Atom（凌动）处理器 CPU 的主要功能及结构，现在我们就来分析其具体的电路原理图设计了。一般而言，一个设计（特别是 X86 平台的设计）的电路图往往都是从处理器（CPU）的连接开始。这一章我们就来说说第三章中模块框图所对应的 X86 平台的嵌入式设计的 CPU 的连接电路。

从前面介绍过的模块框图中可以看到，该处理器（N450 或 D510）已经集成传统意义上的北桥的功能。其主要的接口包括和 DDR2/3 内存连接器连接的 DDR2 memory 总线接口；和南桥相连的 DMI 总线接口；以及连接显示功能的 VGA/LVDS 接口。其原理图的连接其实主要就是这些接口对应的信号与相关元器件的连接。下图是该 Atom 处理器的功能框图。



一，处理器内存 控制器接口信号及连接

处理器内存控制器接口的信号主要就是内存总线的信号。内存总线的信号会在内存章节中做详细介绍。这里就针相关信号连接及电路设计做些分析。下图是该处理器内存控制器的电路连接。



从电路连接上而言，其实这些信号的连接还是比较直接的，大都连接到电路原理图第七页的内存条连接器。以下几点值得关注一下：

－ 内存总线的技术一直在发展。到 DDR2 时，总线的频率已经可以达到 667MHz。DDR2 芯片及相关信号的供电电压是 1.8V 电压。

－ DDR2 内存接口总线主要包括 15 根地址线；片选信号 BS(Bank Selection) 和 CS (Rank Selection)； 64 根数据线分成 8 组，每组 8 位并对应一根 DQS/DQS# (Data Strobe 差分信号)以及 DM (Data Mask)控制信号；控制线 RAS, CAS, WE#，其信号状态组合表示不同的指令；以及时钟信号和时钟使能信号。

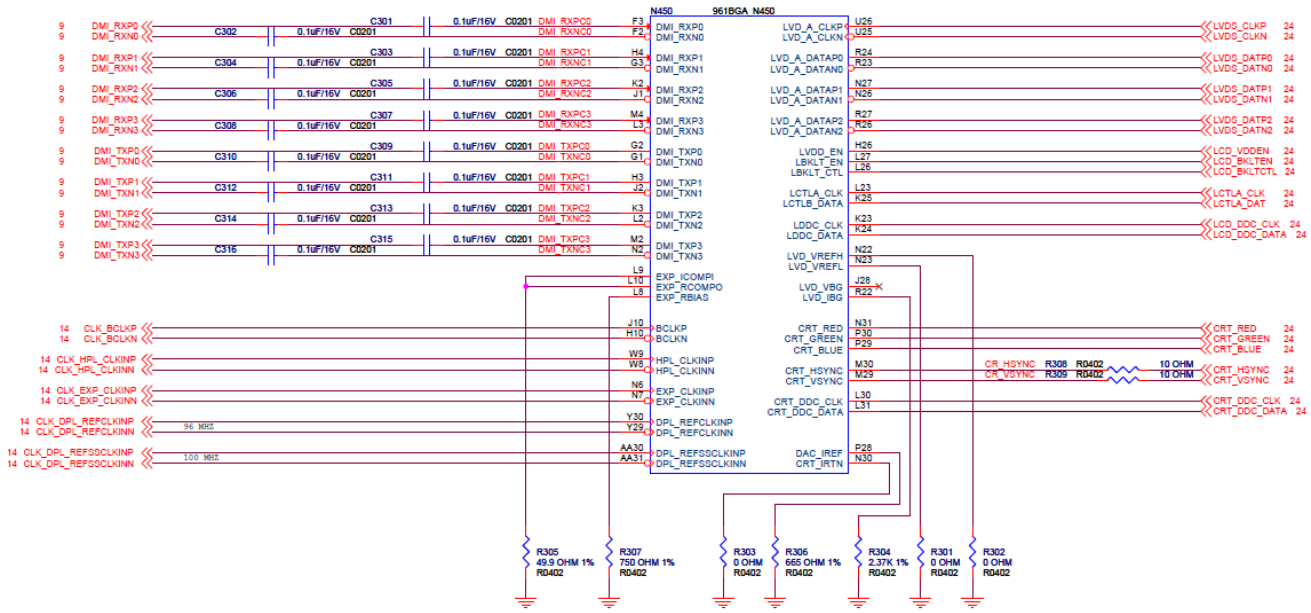
－ 对于 1.8V 电压的 DDR2 接口，其参考电压（DDR_VRFE）是 0.9V。一般直接由 1.8V 通过两个精确的等值电阻分压产生。注意，由于 DDR_VRFE 电压值比较敏感，要求比较精确，因此往往用 1%精度的 49.9ohm 电阻分压，而不用 5%精度的电阻。

－ 内存接口补偿信号，一般遵照 design guideline 的要求去连接，不要随意改变。RPU 通过 80.6ohm 1%精度电阻上拉到 1.8V; RPU 通过 80.6ohm 1%精度电阻下拉连到地（GND）。

课后问题：本设计的只有一个内存连接器，在 DDR2 总线连接中为什么不同的片选信号 CS (Chip Select)连两根(CS0/1)，而 BS (Bank Select)却要连 3 根(BS0/1/2)？

二，处理器 DMI, 显示及时钟信号接口及电路

下图是该处理器 DMI 总线, 显示及时钟信号接口信号的电路连接。这些信号的连接也不复杂，与对应得芯片以及连接器直接相连。



下面对这部分电路进行具体分析。

1，DMI 接口及电路设计

电路左上部分是 DMI 信号连接，全称是 Direct Media Interface。这是英特尔内部开发的用于南北桥之间连接的串行总线。在该设计中，由于北桥的功能已集成到处理器中，因此 DMI 直接连接该（凌动）处理器和 ICH8 南桥。该总线主要包含由南桥发到处理器的 4 组 RX 差分信号对 DMI_RXP/RXN[0:3]，以及由处理器发到南桥的 4 组 TX 信号对 DMI_TXP/TXN[0:3]。处理器的 DMI_RXP/RXN[0:3]对处理器而言是输入信号，连接到南桥端的 DMI 输出信号也就是南桥端的 DMI_TXP/TXN；处

理器的 DMI_TXP/TXN[0:3]对处理器而言是输出信号，连接到南桥端的 DMI 输入信号也就是南桥端的 DMI_RXP/RXN。

EXP_ICOMP/RCOMP 是 DMI 总线的电流和阻抗补偿信号。根据芯片的设计 guideline, 这两个信号连接在一起，通过 50 欧姆电阻连接到地。EXP_RBIAS 是该总线的偏置设定控制，根据设计规范，通过 750 欧姆电阻连接到地。至于为什么要这样连接，这些电阻是如何选定的，就要对芯片内部的电路设计有相当的了解。通常情况，芯片公司不会透露相关细节，所以一般直接照着设计规范的要求连接就可以了。

2，时钟信号及电路设计

由于处理器功能较多，因而需要较多的时钟信号，有核心逻辑电路用的时钟，DMI 接口总线用的时钟，显示接口所使用的时钟信号等等。对于处理器，这些时钟信号是输入信号。它们都由时钟发生器芯片产生，因此电路图左下部分时钟信号都连接到位于 14 页的时钟发生器芯片。

BCLKP/N 是处理器的核心时钟信号，为 166 兆赫兹；

HPL_CLKINP/N 是处理器主设备时钟信号，为 166 兆赫兹；该时钟信号输入到处理器主设备的锁相环电路 (PLL)，通过锁相环电路产生内存控制器所需要的时钟信号以及内存总线接口的时钟信号；

EXP_CLKP/N 是处理器 DMI 接口电路时钟信号，为 100 兆赫兹;

DPL_REFCLKINP/N 是处理器内部 PLL 的参考时钟输入信号，为 96 兆赫兹；该时钟信号输入到处理器内的显示锁相环电路，通过锁相环电路产生显示接口电路；

RFESSCLKINP/N 是展频时钟输入信号，为 100 兆赫兹；有时为了降低显示接口的辐射问题，可以使用展频时钟输入到显示锁相环电路，以后在时钟发生器或显示电路中做进一步分析。

3，显示接口信号及电路设计

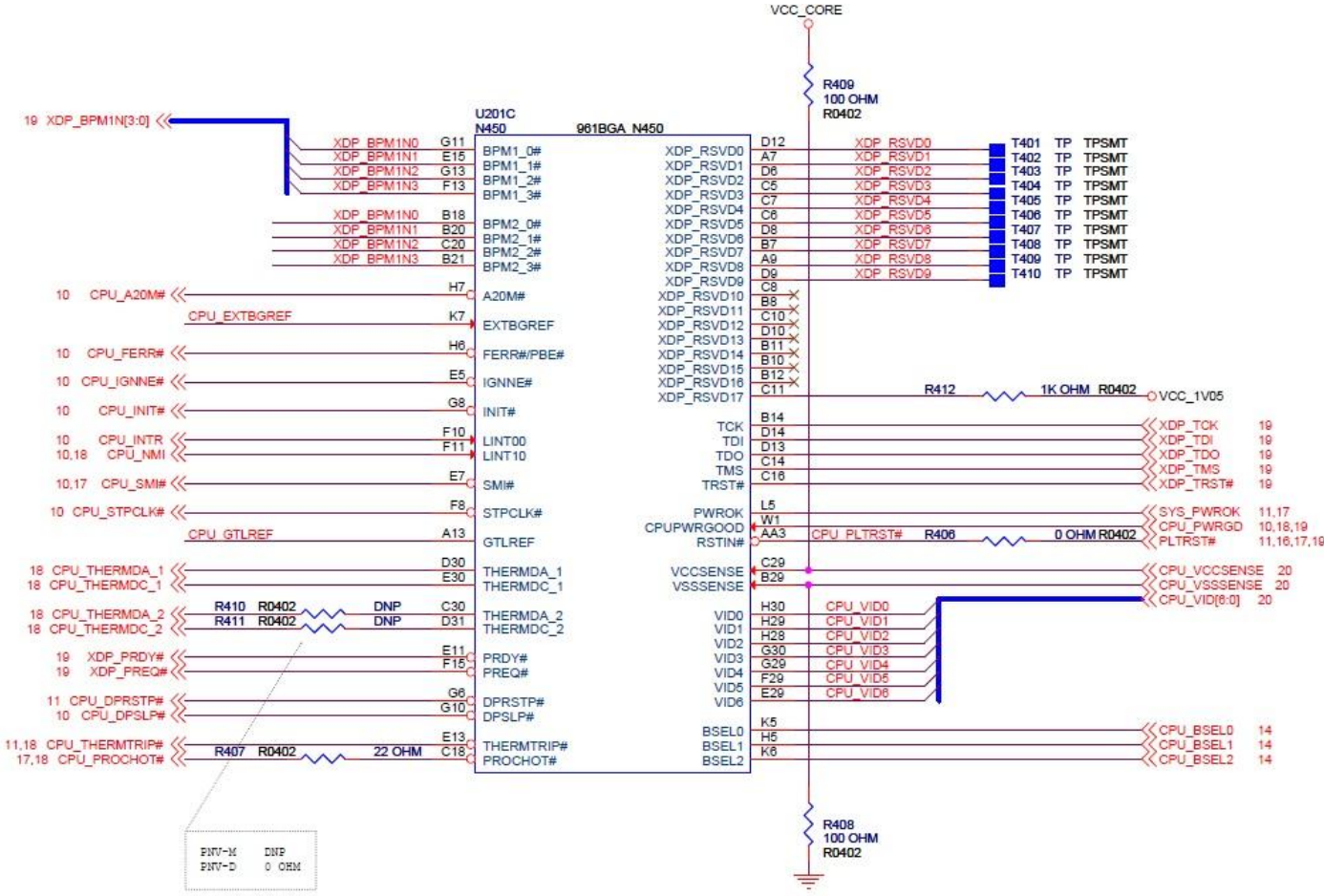
前面电路图的右边部分是显示接口信号的连接。右上部分是数字显示 LVDS 接口，没什么特别好说的，相关信号连接到位于 24 页的对应引脚。LVD_VREFH/L，根据 datasheet，这是两个预留的引脚定义，可以连到地，或者不连接。在本设计中，用 0 欧姆的电阻连到地，这样也预留了调试的空间。万一接地出现问题，可以很方便地把 0 欧姆电阻去掉，而不用更改 PCB 设计。

一　电路图的右下部分是模拟显示 VGA 接口信号，信号也是连接到位于 24 页的连接器。CRT_IRTN 是用于连接电流回路通道，因而根据规范直接连接到地。本设计也是串接 0 欧姆电阻，以备将来调试需要。DCA_IREF，根据规范，通过 665 欧姆 1%精度电阻连接到地。因此就按照规范要求相连接。由此也可以看出，很多引脚的连接一定要参照 datasheet 和设计 guideline 的要求来做。

这部分分享就到这里。课后问题：DMI 总线的信号中串接一个 0.1uf 的电容，该电容有什么作用呢？欢迎一起学习讨论。

三，处理器其他各种接口信号及电路

下图是该处理器其他多种接口信号的电路连接。这些信号的连接虽说大都是与对应的芯片或连接器连接，但功能上却分属不同的接口或总线。



这部分电路具体分析如下：

一　电路左上部分 XDP_BPM 信号以及右边中间的 XDP_TCK/TDI/TDO/TMS/TRST#信号属于 XDP 接口信号。XDP 是

英特尔处理器使用的调试接口。这些信号通过一根电缆线连接到外置的调试器。软件程序开发者可以通过这个调试器以及连接调试器和处理器的 XDP 信号对处理器运行的程序代码进行监控，并可执行一些操作比如设置断点等从而进行程序代码的调试工作。这个接口往往在早期的电路板上放置，供软件开发工程师使用；到后期往往不放，硬件设计者以及最终的用户一般都不用这个接口。

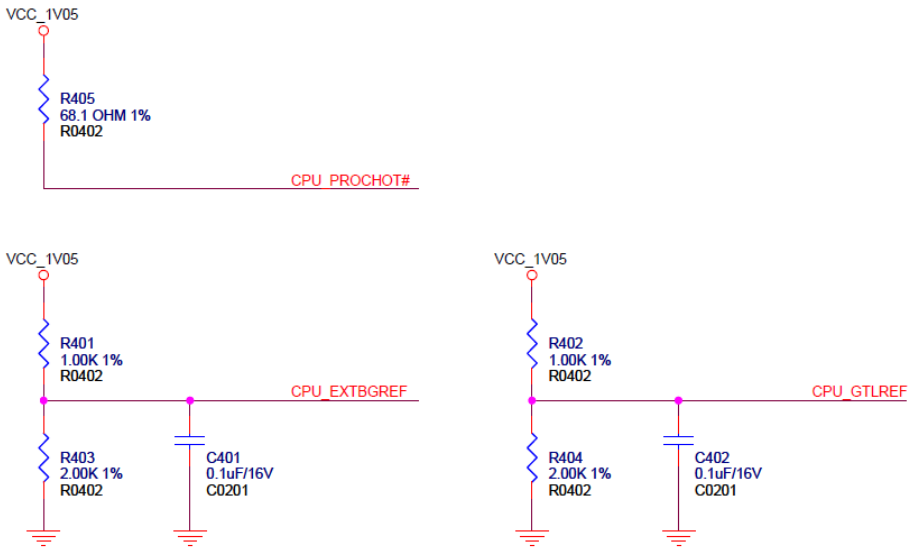
— X86 处理器传统功能信号，包括实模式地址控制信号 A20M#；错误管理信号 FERR#, IGNNE#；重新初始化信号 INIT#；中断控制信号 INTR，NMI，SMI#；控制处理器省电并停止总线使用的 STPCLK#；控制睡眠模式的 DPSLP#和 DPRSTP#信号。这些信号大都直接连接到南桥 ICH 相应的同等功能的信号。除此，还有热敏二极管信号 THERMDA /DC, 连接到第 18 页的外部数模转换芯片，系统可通过此芯片读到处理器的温度。BSEL0-2 是处理器时钟频率控制信号，输出给位于第 14 页的时钟发生器。时钟发生器根据处理器发出的这些信号的状态，从而产生并输出相应的时钟频率给处理器。

— 过温保护信号。当处理器温度达到芯片工作温度范围上限时，处理器会发出 PROCHOT#信号，连接到由系统 EC 控制器或南桥。EC 控制器或南桥可根据设定采取一定措施比如发出 STPCLK#等等，降低处理器功耗从而降低温度。在某些实效情况下，当处理器温度到达极高的一个门限时，处理器会发出 THERMTRIP#，让系统关电，从而保护处理器。THERMTRIP#直接连到南桥和系统 EC 控制器，在极端高温情况下，由南桥或系统 EC 控制器完成系统关电动作。

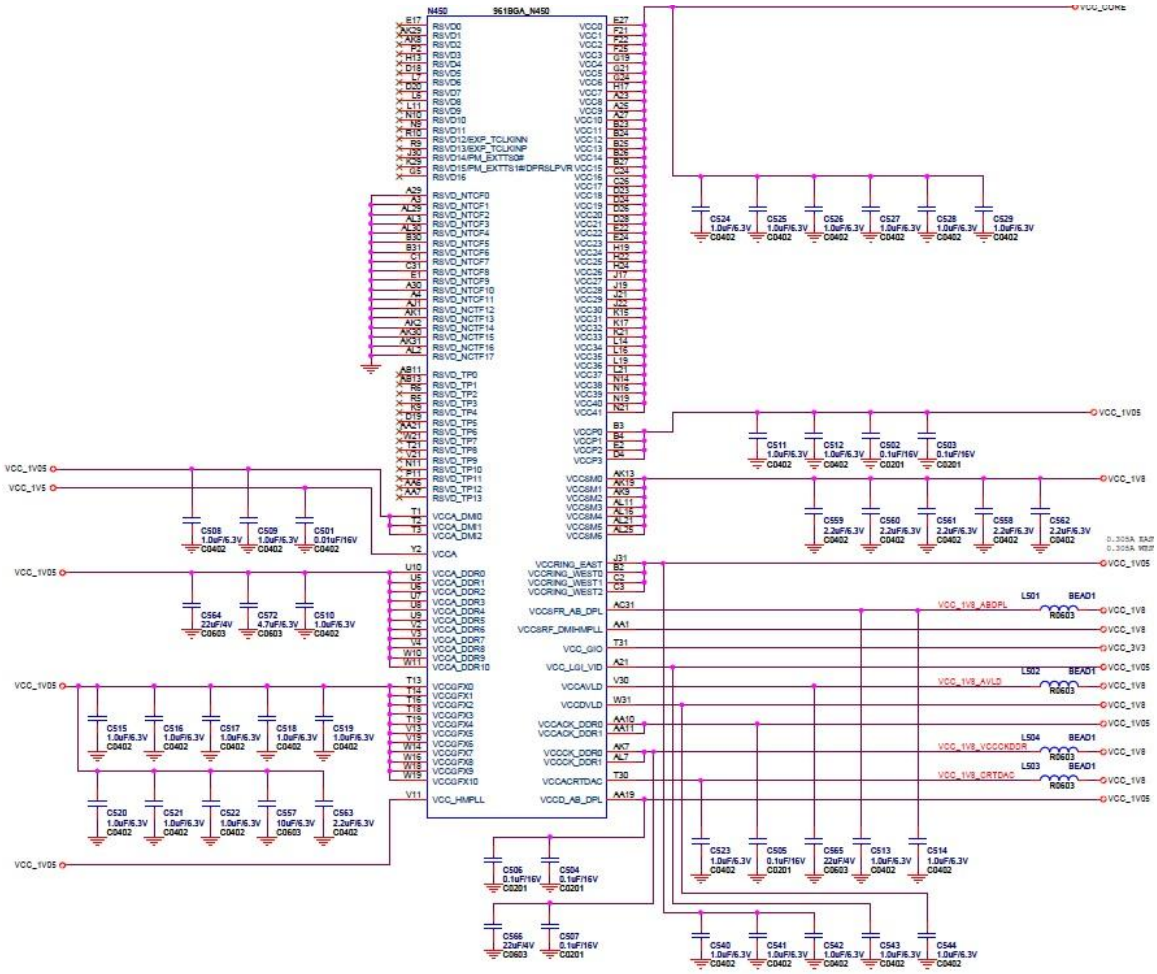
— 电源稳定和复位信号。PWROK 信号用于告诉处理器主电源已经稳定，该信号直接连接到整个系统的电源稳定标志信号 SYSPWROK；CPUPWRGOOD 用于表示系统电源以及时钟信号都已经稳定；RSTIN#是处理器的复位信号，当电源和时钟稳定后，该信号从低到高跳变并保持为高时，处理器复位结束，并跳转到系统 BIOS 入口地址，开始执行指令。该信号直接连到系统 PLT_RST#信号。

— 电源控制信号 VID0-6。这些信号输出到位于第 20 页的 VCC_CORE 开关电源控制器，从而控制该开关电源电路产生正确的电源电压。VCCSENSE 和 VSSSENSE 是在处理其内部和 VCC_CORE 以及地以低阻抗连接在一起，可以为 VCC_CORE 开关电源电路提供一个最接近使用端的干净的反馈电压。

下图是一些辅助电路。PROCHOT#是一个漏极开路信号（open-drain），需要外部上拉。根据设计规范，这里通过 68.1 欧姆电阻上拉到 1.05V。处理器的参考电压通过 1%精度的 1.0K 欧姆和 2.0K 欧姆上下分压产生，连接到处理器的参考电压输入引脚。



下图是该处理器的电源和地引脚及连接。从图中可以看到，该处理器由于支持不同的接口总线功能，也需要多种的电源的支持，包括：VCC_VCORE 的 core 电压，DDR2 内存接口的 1.8V，以及其他接口所需要的 1.5V, 1.05V 等等。



这部分分享就到这里。课后问题：上图中芯片电源引脚与地之间放了很多电容，有些是 1 uf，也有 0.1uf，这些电容有什么作用呢？欢迎一起学习讨论。

第六章　内存(memory)芯片功能及信号

内存（英文叫 memory）在现代计算机系统中的应用非常广泛，可以说不管是嵌入式设计还是 PC/服务器/存储器/网络交换机系统中是不可或缺的关键组件。以前发布的计算机系统架构演变，以及《硬件设计新手入门宝典》第一部中的计算机系统架构部分都对内存的作用及技术发展做过一些介绍。

一， 内存功能及应用介绍

在《硬件设计新手入门宝典》第一部计算机系统架构文中曾介绍过，计算机的组成结构中，有一个很重要的部分，就是存储器。存储器是用来存储程序和数据的部件，对于计算机来说，有了存储器，才有记忆功能，才能保证正常工作。存储器的种类很多，按其用途可分为主存储器和辅助存储器，主存储器又称内存储器（简称内存，港台称之为记忆体）。

内存又称主存，是 CPU 能直接寻址的存储空间，由半导体器件制成。内存的特点是存取速率快。内存是电脑中的主要部件，它是相对于外存而言的。我们平常使用的程序，如 Windows 操作系统、打字软件、游戏软件等，一般都是安装在硬盘等外存上的，但仅此是不能使用其功能的，必须把它们调入内存中运行，才能真正使用其功能，我们平时输入一段文字，或玩一个游戏，其实都是在内存中进行的。就好比在一个书房里，存放书籍的书架和书柜相当于电脑的外存，而我们工作的办公桌就是内存。通常我们把要永久保存的、大量的数据存储在外存上，而把一些临时的或少量的数据和程序放在内存上，当然内存的好坏会直接影响电脑的运行速度。

内存就是暂时存储程序以及数据的地方，比如当我们在使用 Word 处理文稿时，当你在键盘上敲入字符时，它就被存入内存中，当你选择存盘时，内存中的数据才会被存入硬（磁）盘。在进一步理解它之前，还应认识一下它的物理概念。

内存一般采用半导体存储单元，包括随机存储器（RAM），只读存储器（ROM），以及高速缓存（CACHE）。只不过因为 RAM 是其中最重要的存储器。（synchronous）SDRAM 同步动态随机存取存储器：SDRAM 为 168 脚，这是 PENTIUM 及以上机型使用的内存。SDRAM 将 CPU 与 RAM 通过一个相同的时钟锁在一起，使 CPU 和 RAM 能够共享一个时钟周期，以相同的速度同步工作，每一个时钟脉冲的上升沿便开始传递数据，速度比 EDO 内存提高 50%。DDR（DOUBLE DATA RATE）RAM ：SDRAM 的更新换代产品，他允许在时钟脉冲的上升沿和下降沿传输数据，这样不需要提高时钟的频率就能加倍提高 SDRAM 的速度。随着技术的发展，内存已历经 DDR2，DDR3，现在已发展到 DDR4。

1，物理存储器和地址空间

提到内存，不得不说说地址和寻址相关的一些概念。物理存储器和存储地址空间是两个不同的概念。但是由于这两者有十分密切的关系，而且两者都用 B、KB、MB、GB 来度量其容量大小，因此容易产生认识上的混淆。初学者弄清这两个不同的概念，有助于进一步认识内存储器和用好内存储器。

物理存储器是指实际存在的具体存储器芯片。如主板上装插的内存条和装载有系统的 BIOS 的 ROM 芯片，显示卡上的显示 RAM 芯片和装载显示 BIOS 的 ROM 芯片，以及各种适配卡上的 RAM 芯片和 ROM 芯片都是物理存储器。

存储地址空间是指对存储器编码（编码地址）的范围。所谓编码就是对每一个物理存储单元（一个字节）分配一个号码，通常叫作“编址”。分配一个号码给一个存储单元的目的是为

了便于找到它，完成数据的读写，这就是所谓的“寻址”（所以，有人也把地址空间称为寻址空间）。

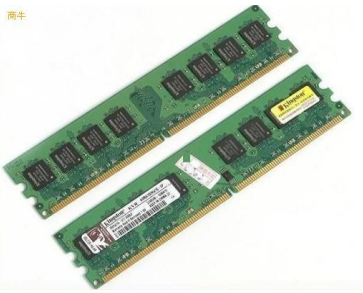
地址空间的大小和物理存储器的大小并不一定相等。举个例子来说明这个问题：某层楼共有 17 个房间，其编号为 801～817。这 17 个房间是物理的，而其地址空间采用了三位编码，其范围是 800～899 共 100 个地址，可见地址空间是大于实际房间数量的。

对于 386 以上档次的微机，其地址总线为 32 位，因此地址空间可达 2 的 32 次方，即 4GB。（虽然如此，但是我们一般使用的一些操作系统例如 windows xp、却最多只能识别或者使用 3.25G 的内存，64 位的操作系统能识别并使用 4G 和 4G 以上的的内存。

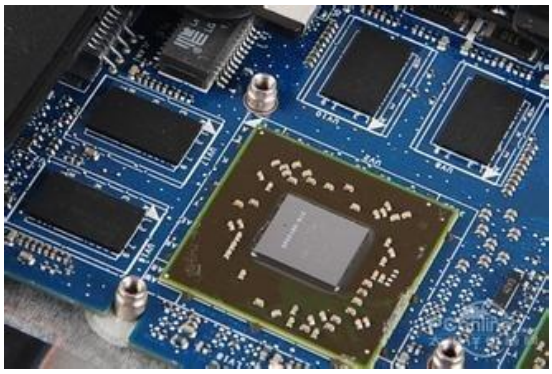
2， 内存的应用

在计算机系统中，内存有两种不同的应用形式：分立内存条和集成设计。

对于 PC/服务器/存储器等复杂计算机系统而言，由于内存的容量比较大，一般都采用分立内存条的应用形式。这种应用主板上设计有内存插槽（如下图）。内存总线信号从内存控制起连接到内存插槽。内存条（如图右）则是由有相关内存厂家设计制造，其上集成多颗内存颗粒，以组成不同容量大小的内存条。内存总线信号在内存条上从金手指接口连接到不同的内存颗粒。内存条插到主板的内存插槽中，从而内存条上金手指信号与内存插槽的引脚信号就连接在一起，从而实现主板上内存控制器芯片到内存条内存颗粒的信号的通路。

对于小型的嵌入式应用而言，所许的内存容量也比较小，为了节约成本，往往没有必要采取分立内存条的应用形式，而是直接把内存颗粒设计到主板上。内存信号直接从主板上的内存控制器连接到主板上的内存颗粒。



二， 内存颗粒（芯片）配置及容量

内存颗粒（实际就是内存芯片）是内存的基本器件。无论是分立的内存条，还是电路板上集成的内存设计，都是由一颗或多颗内存颗粒组成。内存颗粒/芯片的供应商有不同厂家，比如美光科技（Micron），三星(Samsung)，英飞凌(Infineon)，现代（Hynix）等等。下面就以美光的 512Mbit DDR2 内存芯片为例把内存芯片技术做些详细讲解。

DDR2 SDRAM

MT47H128M4 – 32 Meg x 4 x 4 banks
MT47H64M8 – 16 Meg x 8 x 4 banks
MT47H32M16 – 8 Meg x 16 x 4 banks

1，内存芯片的寻址及配置

内存芯片内部存储空间由多个 bank 组成，我们这里举例的美光 512Mbit DDR2 内存芯片就有 4 个 bank，每个 bank 的存储空间是 512 Mbit/4=128 Mbit。每个 bank 就是一个 DRAM 存储阵列，包含很多的存储格点。Bank 里面的存储格点是通过行地址和列地址来定位寻址的。

为了更好地理解上面讲述的内存芯片寻址原理，我们打个形象的比喻。这就好比到一个书库里面找书，书库有一排排书架，每个书架又有好多书格，每个书格里面存放一定数量的图书。上面说的 Bank 就好比这里的一排书架，Bank 里面的存储格点就好比这里书架中的一个书格。找书的时候，得首先知道是那个书架。找到书架后，根据一排书架上行数和列数，找到相应的书格，然后到书格中找到需要的图书。内存的寻址非常类似，首先得通过 Bank 地址找到对应的 Bank，然后通过行地址和列地址寻址到 Bank 内的存储格点，从而就可以进行读写操作了。

内存芯片上所连接的数据信号数量也就是内存芯片的数据位数。不同内存芯片的数据位数各不相同，有些芯片是 4 位，

有些是 8 位，也有 16 位。 不同数据位数也就导致内存内部的配置不尽相同。前面说的 Bank 里面的存储格点，每个格点里面就包含了与内存芯片数据位数等同的存储比特（bit）数。当存储格点通过行列地址被定位到时，就会输出其所有的数据比特到数据信号线上（读指令）或输入数据信号线（写指令）。

根据 Bank 的空间和内存芯片的数据位数，就可计算出 Bank 内的存储格点数，从而进一步确定寻址所需要的行地址信号数和列地址信号数。

前面说过我们这里举例的美光 512Mbit DDR2 内存芯片其 bank 的存储空间是 128 Mbit。如果其数据位是 8 位，则其存储格点数为 128 M/8=16M，其配置(configuration)就是 16M x 8 (位) x 4 banks，如下图的第三列所示。由于 bank 数量为 4，学过二进制的都应该知道两根信号 BA[0]/BA[1]就可以定位 bank 地址(band address)。对于 16M 的存储格点，行地址（Row Address）由 14 根地址线 A[13:0]寻址，共有 16K 行。则列数量为 16M/16K=1K，需要用到 10 根地址线 A[9:0]来定位列地址。

Parameter	128 Meg x 4	64 Meg x 8	32 Meg x 16
Configuration	32 Meg x 4 x 4 banks	16 Meg x 8 x 4 banks	8 Meg x 16 x 4 banks
Refresh count	8K	8K	8K
Row address	A[13:0] (16K)	A[13:0] (16K)	A[12:0] (8K)
Bank address	BA[1:0] (4)	BA[1:0] (4)	BA[1:0] (4)
Column address	A[11, 9:0] (2K)	A[9:0] (1K)	A[9:0] (1K)

如果内存芯片位数为 4 位，该芯片的一个 bank 的存储空间是 128 Mbit，则一个 bank 的存储格点数为 128M/4=32M，所以其配置为 32M x 4 (位) x 4 banks，如上图第二列所示。bank 数量 4，同样由两根信号 BA[0]/BA[1]可以定位；32M 的

存储格点，行地址由 14 根地址线寻址，共有 16K 行；则列数量为 32M/16K=2K，需要用到 11 根地址线来定位列地址。

如果内存芯片位数为 16 位，该芯片的一个 bank 的存储空间是 128 Mbit，则一个 bank 的存储格点数为 128M/16=8M，所以其配置为 8M x 16 (位) x 4 banks，如上图第四列所示。bank 数量 4，同样由两根信号 BA[0]/BA[1]可以定位；8M 的存储格点，行地址由 13 根地址线寻址，共有 8K 行；则列数量为 8M/8K=1K，需要用到 10 根地址线来定位列地址。

2，内存芯片容量计算和选择

上述这部分内容应当搞清楚，对于常常把内存芯片集成到电路板上的嵌入式设计而言，往往需要根据这些内容来计算内存颗粒的数量及定义相关寻址信号的连接。

举个例子：如果你的嵌入式系统设计需要 256MByte 的内存(注意这里是 Byte 不是 bit)，控制处理器芯片内存接口数据是 8 位的，Rank 选择信号有两个 CS0#/CS1#。这种应用应当选什么样的内存芯片呢？

由于处理器端的数据位是 8 位，内存芯片的数据位应避免大过处理器端的数据位。所以选择的内存芯片是 8 位或 4 位

- 如果要选择 4 位的内存芯片

  - 由于有两根 Rank 选择信号，可以支持最大 2 个 Rank。处理器端的数据位是 8 位，而内存芯片也是 4 位，就是说每个 Rank 可支持 2 个芯片（处理器端的数据位/内存芯片数据位=2），两个 Rank 可以支持 4 个芯片。每个芯片容量是 256MB x 8/4 = 512Mb, 也就是说应该选 512Mb /4 位（32Mx4x4banks）的内存芯片
  - 也可以选择 1 个 Rank，这样也就可以用总共 2 个内存芯片，每个芯片容量是 256MB x 8/2 = 1Gb。也就是说应该选 1Gb/4 位（64Mx4x4banks）的内存芯片

- 如果要选择 8 位的内存芯片
  - 由于有两根 Rank 选择信号，可以支持最大 2 个 Rank。处理器端的数据位是 8 位，而内存芯片也是 8 位，就是说每个 Rank 只能支持 1 个芯片（处理器端的数据位/内存芯片数据位=1），两个 Rank 可以支持 2 个芯片。每个芯片容量是 256MB x 8/2 = 1Gb，也就是说应该选 1Gb/8 位（32Mx8x4banks）的内存芯片
  - 当然也可以用 1 个 Rank，也只能用一个内存芯片，其容量较大 2Gb/8，不知到能不能找到这样的芯片

当然内存芯片的选择除了计算其容量，数据位数等参数外，还的看其它的一些特性要求，比如控制处理器端支持什么内存技术（DDR, DDR2, DDR3?），内存芯片也要与之匹配。还有比如电路板空间的大小。用上面的例子来说，如果电路板上

没空间放 4 颗内存芯片，只能放 2 颗，那么就只能选 1Gb/4 位或 1Gb/8 位的芯片了。

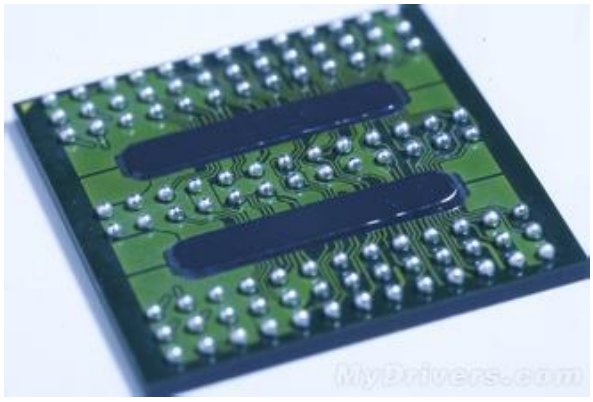
三， 内存芯片封装及信号

1，内存芯片封装

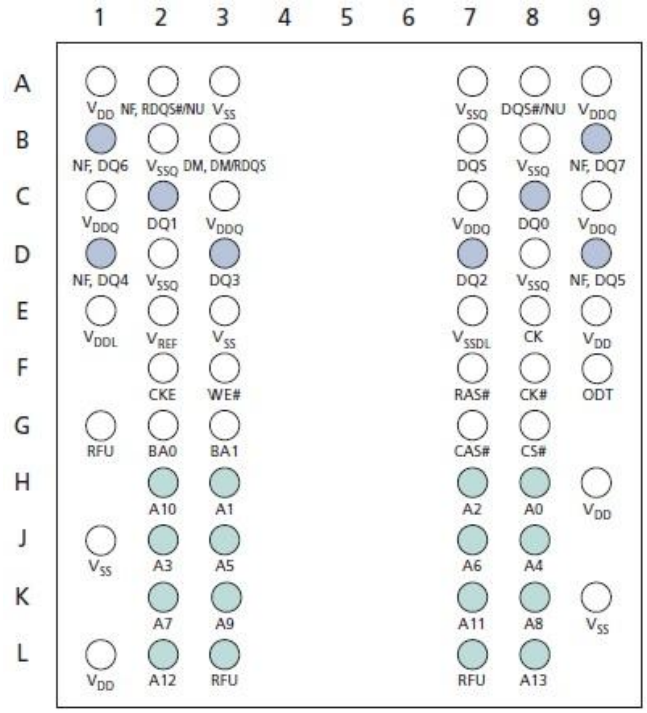
内存的技术，存储容量，数据传输速度不断在进步，生产制造工艺也不断在更新。内存芯片的封装也经历了很大的变化。上图是传统的 TSOP 薄型小尺寸封装（引脚分布在内存芯片的两边，引脚之间的间距很小，采用 SMT 技术（表面安装技术）直接附着在 PCB 板的表面。

现在的内存芯片多采用球栅阵列封装，简称 BGA(Ball Grid Array Package)。BGA 封装的 I/O 端子以圆形或柱状焊点按阵列形式分布在封装下面，如下图所示。采用 BGA 技术封装的内存，可以使内存在体积不变的情况下内存容量提高两到三倍，BGA 与 TSOP 相比，具有更小的体积，更好的散热性能和电性能。BGA 封装技术使每平方英寸的存储量有了很大提升，

采用 BGA 封装技术的内存产品在相同容量下，体积只有 TSOP 封装的三分之一；另外，与传统 TSOP 封装方式相比，BGA 封装方式有更加快速和有效的散热途径。



前一节中举例的美光 512Mbit DDR2 内存芯片就采用一种 BGA 封装叫 FBGA（Fine-Pitch Ball Grid Array）：细间距球栅阵列。下图就是芯片底部焊点及其所对应信号的分布图。



这些球形焊点所对应的信号就是内存芯片的接口信号。嵌入式处理器或桥内部的内存控制器就是通过这些接口信号对内存芯片进行读写操作。下面就对这些接口信号的功能作些解释，了解这些内容后，内存的信号连接及电路设计就更好理解。

2，内存芯片的接口信号

总体而言，内存芯片的接口信号按照功能来说主要分为三大类：地址信号，数据信号，和控制信号。内存芯片通过这些内存总线信号与内存控制器 host 端连接在一起。

地址信号：顾名思义，就是用来寻找对数据进行读或写操作的具体位置的信号。一个内存芯片内部有较大的存储空间，要能到读出想要的数据或正确写入数据，必须得先告诉内存想要读出的数据或要写入的数据位于什么位置。就像寄快递一样，如果你有一个包裹要请快递公司寄出去，你必须要告诉快递公司你自己的位置（这样快递公司可以派收件员上门收取包裹）以及你要寄到的目标地址（快递公司根据这个地址送包裹上门）。前面举例的美光 512Mbit DDR2 内存芯片，

- 4/8 位的内存芯片使用 14 根地址信号 A[0:13] (A0，A1，…，A13)
- 16 位的内存芯片使用 13 根地址信号 A[0:12] (A0，A1，…，A12)
- 内存芯片内部 Bank 的寻址信号 BA0，BA1

前面介绍内存寻址原理时及内存芯片结构都曾提到过：内存芯片内部是存储阵列，包含很多的存储格点，需要通过行和列进行地址确认。那么上面的地址信号是如何进行行列地址寻址的呢？我们先来看看一个会议室安排的例子。假设有两个团队A和B都有开会的需求，因而都要用到会议室。我们当然可以申请两个会议室，分别归属A和B，这样这两个团队都拥有各自的会议室，可以随时使用，没有冲突；还有另外一种办法，就是把A和B团队的会议安排在不同的时间段，这样他们都可以用同一个会议室，因而只用安排一个会议室给他们就可以。考虑到两个团队不可能一直开会，错开开会基本没什么问题，所以后一种方式无疑更为有效经济。这里地址信号的使用也很类似：同一组信号错开使用，在行寻址阶段表示行地址信号，在列寻址阶段表示列地址信号。一个完整的寻址故而包含行寻址和列寻址阶段，虽然寻址的时间增加一点，但极少降低了地址信号的数量，从而降低了电路板设计的复杂度。具体的寻址过程在前述的寻址原理的章节中有过讲述，就不再赘述。

地址信是单向的，由内存控制其传送到内存芯片。因此对于内存芯片而言，地址信号是输入信号；而对于内存控制器而言，地址信号则是输出信号。这其实用前面说的快递过程来解释更清楚一点。快递员可以送包裹上门，也可以上门取包裹（从而送到包裹寄去的目的地）。但不管送包裹也好，取快递也好，都是快递员找到相应的地址完成后续操作，也就是说快递员是单向的找到客户（送到的或者取件的对象）的地址。内存总线的工作过程也累似，不管读数据还是写数据，地址都由内存控制器 host 端发出（相当于快递员）。

数据信号：就是一组或多组用来传送数据的信号，包括传送数据信号及数据锁存信号。这些信号是双向的，也就是说数据传送方向可以从内存控制器到内存芯片，也可以反过来从内存芯片到内存控制器。从内存控制器到内存芯片的数据传输就是数据写操作，内存控制器根据 CPU 指令把数据传送到内存芯片，从而写入到内存芯片相应的地址中；从内存芯片到内存控制器的数据传输就是数据读操作，内存控制器根据 CPU 指令告诉内存芯片要读某地址的数据，内存芯片就输出相应数据到数据总线上并传送到内存控制器。

如果用快递来类比的话，可能会理解起来更容易一些。快递员和一户家庭之间是双向的关系。如果有人寄包裹到一户人家，

那么快递员就会传送包裹到这户人家（类似上面的写操作）；如果这户人家有包裹寄出去，快递员就会上门提取包裹（类似上面的读操作）。上述内存控制器与内存芯片之间的数据传送关系其实与这里的快递员和一户家庭之间包裹传送关系非常类似。

内存芯片数据信号的数量由其位数来决定。以前面讲过的美光 512Mbit DDR2 内存芯片为例：

- 32M/16 位：有 16 根数据信号 DQ[15：0]
- 64M/8 位：有 8 根数据信号 DQ[7：0]
- 128M/4 位：有 4 根数据信号 DQ[3：0]

除了传送数据的内存数据信号外，数据总线上还有数据锁存信号差分信号对，也就是通常所说的 DQS 和 DQS#。这组信号与上面讲过的数据信号完全同向，也就是说该信号也是双向信号。

- 读数据时，该信号由内存芯片输出，输入到内存控制器 host 端
- 写数据时，该信号由内存控制 Host 输出，输入到内存芯片端

数据锁存信号，其主要作用就是要把数据信号锁存助，从而保证数据信号能够传输到终端，并被终端正确识别。说得形象一点，这就好比快递员用来把包裹固定在电瓶车上的绳索。如下图所示，这么多包裹（数据）在运输过程（传输）当中如果没有绳索加以固定，就很容易丢失，或掉落损坏。内存数据也是如此，为保证数据正确传输，并被终端识别，就使用数据锁存信号，把数据信号线上的数据状态锁存下来，不丢失，也不损坏。

上面的图中运输得包裹数量有点夸张了。实际，当数量达到一定程度时，未降低运输过程中的风险，最好分成不同的车来运

输。这样，每部车上都会用自己的固定绳索。内存芯片的数据传输也是如此：当数据信号数量达到一定程度时，就要分成不同的数据信号组，每组信号线有单独的数据锁存信号。比如：

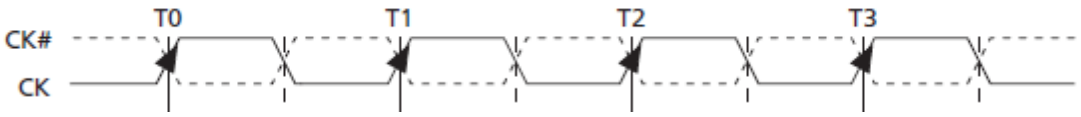
- 128M/4 位：有 4 根数据信号 DQ[3：0]，使用一组锁存信号差分组
- 64M/8 位：有 8 根数据信号 DQ[7：0]，使用一组锁存信号差分组
- 32M/16 位：有 16 根数据信号 DQ[15：0]，使用 2 组锁存信号差分组
    - UDQS, UDQS#，负责高 8 位数据信号的锁存
    - LDQS, LDQS#，负责低 8 位数据信号的锁存

控制信号：就是一系列传输内存总线及内存芯片操作指令的信号，包括时钟信号 CK/CK#，时钟使能信号 CKE，片选信号 CS#，数据屏蔽信号 DM，LDM，UDM，内存芯片内部终端电阻使能信号 ODT，行地址信号 RAS#，列地址信号 CAS#，以及读写控制信号 WE#。

前面讲过内存总线上有地址信号和数据信号，那么为什么还需要控制信号呢？前面也用送快递过程举例，我们接着用这个例子来解释。当快递员要送一个包裹（数据）到某个地址时，他会把包裹装在自己电瓶车（数据总线）上。但是光有包裹，地址信息以及电瓶车还不能把包裹送到。那还需要什么呢？电瓶车要发动才能跑起来；快递员必须正确控制电瓶车的方向确保沿着正确方向前进；到了目的地后快递员还需要敲门并确认正

确送达对象后才能完成包裹送达任务，这些都是控制过程，包括对电瓶车以及包裹的控制和操作。同样内存总线的数据传输光有地址线，数据线，数据等是不够的，还需要有针对总线以及地址/数据信号的控制，而这些控制指令的传达就是通过控制信号来实现的。下面我们就解释一下这些信号的功能。

- CK/CK#是差分时钟信号，对内存芯片而言，这是输入信号。所谓时钟差分信号，就是两个相位完全反向的时钟信号，如下图所示。两个信号频率完全相同，只不过 CK#与 CK 相位正好相反。



大家都知道，在数字电路设计中，时钟信号可以说整个电路系统运行的基础。数字电路逻辑器件及芯片都参考时钟信号而形成其内部状态机的状态转换，从而最终实现一定功能。没有时钟信号，数字逻辑电路就没有头绪，状态机运行就会停顿，功能当然就不能实现。内存芯片也是如此，所有地址信号和控制信号，其逻辑状态的检测都是在 CK 时钟的上升沿以及 CK#的下降沿的交叉点完成。

- CKE 是时钟使能信号。当 CKE 为高时，内存芯片内部的基于时钟运行的逻辑电路被激活；当 CK 为低时，相关基于时钟运行逻辑电路则被冻结。这是什么意思呢？当内存芯片正常工作进行读写操作时，前面说过这些逻辑状态转换从而实现一定功能都要参考时钟信号，这时 CKE

信号必须保持为高。当 CKE 信号为低时，芯片内部逻辑器件就失去时钟信号的参考，因为状态机不能正常转换，相应功能也不能实现，芯片的功耗当然就显著下降，这也就是省电模式。因此 CKE 一个重要用途就是控制内存芯片进入和退出省电模式。

- CS#，就是常说的片选信号。当 CS#为低时，内存芯片内部的指令解码器就被使能或激活，能过对总线指令进行解码并完成相关操作。当 CS#为高时，指令解码器就被冻结，内存芯片就不会对总线的指令进行解码，也就不会承担后续的操作指令，这也就是我们通常说的该内存芯片没有被片选选中。在多 rank 内存系统或内存条设计中，CS# 也被用来作为 rank 的选择信号。

- 数据屏蔽信号 DM，LDM，UDM，这些信号是针对芯片输入数据（也就是写数据）的屏蔽信号，它们对芯片来说也是输入信号。在读数据的场合（也就是内存芯片输出数据的情况），不会用到这些个屏蔽信号。为什么要用这些数据屏蔽信号呢？还是来看看以前说的送快递的例子。如果快递员接到 5 个包裹要送到同一个小区的 5 户不同人家，到了小区后，快递员与这 5 户人家联系一下，3 户有人在，2 户没人在家。不难理解，快递员当然会把有人在家那 3 户的包裹给送过去；但没人在家那两户的包裹就不会送过去（因为没人接受），可以做个无人在家接受标识，重新找时间发送。内存芯片写入数据也可能会遇到类似情况。有些芯片是位数较多，比如可能有

两个 byte。在写入时有很多情况只写入一个 byte 到某个地址，但为提高内存总线的利用率，内存控制器往往会往所有数据位上写入数据。也就是说会出现数据位上有 2 个 byte，但只需要把其中一个写到相关地址的情况，这时控制器会把无效写入数据 byte 所对应的数据屏蔽信号置位（为高）。这样当内存芯片检测到相关数据屏蔽信号为高时，就会忽略到这一数据，从而保证数据写入的正确操作。LDM 是针对低位数据的屏蔽信号，而 UDM 则是针对高位数据的屏蔽信号。

- ODT：是内存芯片内部终端电阻的使能信号。有不少电路板在内存芯片或内存插槽后会加终端电阻，所以并不是所有系统都需要使用芯片内部终端电阻。如果要使用，ODT 信号需要被置位，这对内存芯片业是输入信号。

- RAS#/CAS#/WE#，是总线的操作指令信号，对于内存芯片而言，是输入信号。如下图所示，这些信号的不同状态的组合表示不同的操作指令。这里就不针对这份真值表一个一个去讲，只是简述一下最基本的两个操作。当 RAS#为高，CAS#为低，WE#为低，这种状态组合表示数据写操作；当 RAS#为高，CAS#为低，WE#为高，这种状态组合表示数据读操作。当然前提条件就是 CKE 信号为高，基于时钟运行的逻辑电路要被激活，而且 CS#信号要为低，也就是芯片的片选信号要被选中。

Function	CKE		CS#	RAS#	CAS#	WE#
	Previous Cycle	Current Cycle				
LOAD MODE	H	H	L	L	L	L
REFRESH	H	H	L	L	L	H
SELF REFRESH entry	H	L	L	L	L	H
SELF REFRESH exit	L	H	H	X	X	X
			L	H	H	H
Single bank PRECHARGE	H	H	L	L	H	L
All banks PRECHARGE	H	H	L	L	H	L
Bank ACTIVATE	H	H	L	L	H	H
WRITE	H	H	L	H	L	L
WRITE with auto precharge	H	H	L	H	L	L
READ	H	H	L	H	L	H
READ with auto precharge	H	H	L	H	L	H

其它信号：除了上面讲的数据，地址及控制信号，内存芯片剩下的引脚连接主要是电源信号和地信号了。DDR2 的内存芯片的电源电压是 1.8V，因此内存的芯片上有不少连接 1.8V 电源的引脚。考虑到电源完整性及抗干扰性的要求，内存芯片把 1.8V 电源引脚分成几类：

- VDD，主要是给芯片内部地址/控制信号接口及主要控制逻辑电路提供电源。
- VDDQ，主要是给数据及锁存信号接口及逻辑电路提供电源。

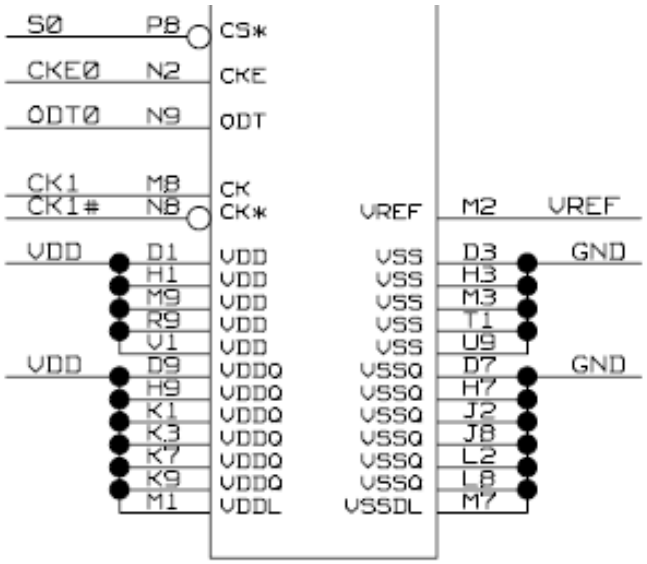
- VDLL，主要是给内存芯片内部的 DLL（延迟锁相环电路）提供电源。DLL（延迟锁相环电路）电路主要是控制内存芯片数据输出（也就是写操作）时的数据信号和锁存信号的时序。

同样，由于信号完整性和抗干扰的考虑，地线引脚也分成几类：

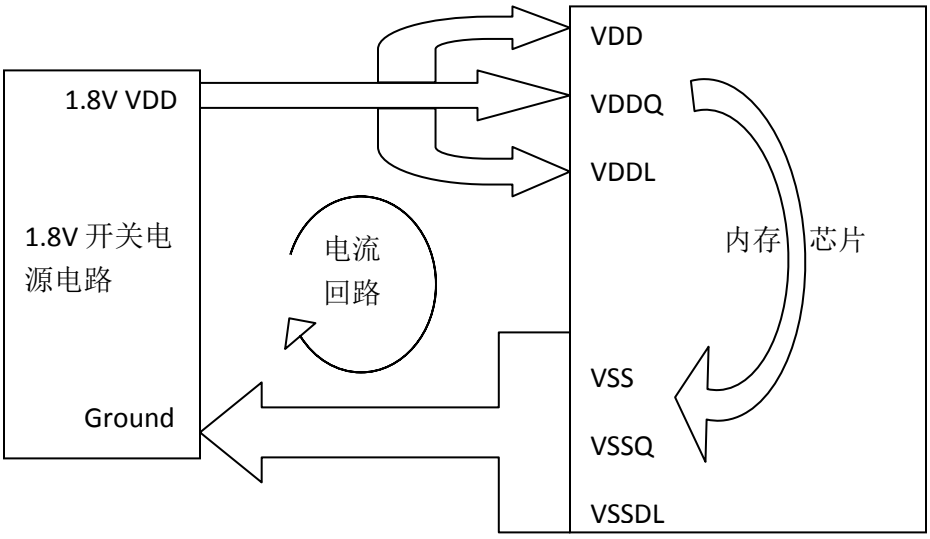
- VSS，主要是给芯片内部地址/控制信号接口及主要控制逻辑电路提供地回路连接。
- VSSQ，主要是给数据及锁存信号接口及逻辑电路提供地回路连接。
- VSSDL，主要是给内存芯片内部的 DLL（延迟锁相环电路）提供地回路。

VDD/VSS， VDDQ/VSSQ 都是有多个引脚连接到这些信号；VDDL/VSSDL 则是一个电源和一个地引脚，如下图所示。那么为什么前面两组电源和地需要多个引脚呢？这两组电源/地分别是提供电源回路给地址/控制总线接口和相关逻辑电路，以及数据信号接口和相关逻辑电路。这些接口及逻辑电路对应的信号较多，因而所需要的逻辑电路状态机也就比较多一些，需要的电源当然就多一些。由于芯片每个引脚能够通过的电流是有限的，当某个功能电路需要较多电流时，就需要多个电源/地引脚从而允许足够的电流流过。这就好比一部车子限到淤泥坑中，需要人推出来。这时靠一个人的力量是不够的，很难推动。多个人一起联合使力，推出来就轻而易举了。芯片的电源

引脚只有足够多，才能给芯片提供足够的电流，从而保证芯片的正常工作。

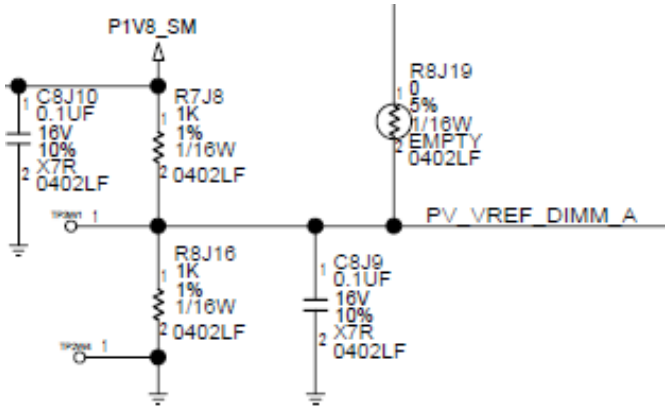


对于芯片而言，电流是从电源引脚流入，从地引脚流出，从而形成完整的电流回路，如下图所示。因此不仅是电源引脚及输入铜箔过孔需要考虑电流通过能力；地引脚数量，PCB 上地铜箔宽度及地过孔也要有相同的考虑和设计。



参考电压 VRFE: 是内存总线信号的参考电压，电压值是 1/2*VDD，对于 DDR2 内存芯片来说，就是 0.9V。这个参考电压有什么作用呢？前面说过内存总线有不同的信号，这些信号都有高和低两个状态，那么内存芯片和内存控制器是如何判断这些信号的状态呢？这其实就是参考电压发挥作用的地方。举个例子，学生都要考试的，如果你问一个学生考得怎么样，学生回答“还可以”，你并不能清晰判断该学生考试的成绩怎么样。但如果该学生告诉你考了 90 分，你马上就会得出明确的结论该学生的考试成绩挺好。因为我们有个约定俗成的参考标准：60 分-及格线。90 分远高于及格线标准，应该是个好结果。参考电压 VRFE 就相当于上述例子中的及格标准线。当 DDR2 总线信号的实际电压高于该参考电压（0.9V）时，芯片就认为信号为逻辑“1”（高）；当 DDR2 总线信号的实际电压低于该参考电压（0.9V）时，芯片就认为信号为逻辑“0”（低）。

所以参考电压对于总线的正常工作非常关键，该电压值要求非常精确，而且应当避免干扰。一般该参考电压由两个等值精密电阻（1%精度）从 1.8V 电源分压产生，如下图。



到这里，内存芯片的信号讲解就告一段落了，DDR2 芯片上所使用到所有信号都涉及了。当然除了上述实际被用到信号及引脚，DDR2 内存芯片上还有些引脚没有被用到：有些引脚没有功能，比如 8 位的芯片中用到的高 4 位数据引脚在 4 位的芯片上就没有实际功能；有些引脚是预留为未来扩展用，比如 Bank Address BA2 以及一些高位的地址信号等等。

四，常用 DIMM 内存条类型

后面会有章节详细讲解内存条的电路设计。这里提到一些常用关于内存条的名词，比如 non-buffer，non-ECC，SO-DIMM，不妨对这些词语作些解释，以帮助大家理解。

DIMM

DIMM ,英文全称是 dual in-line memory module，翻译成中文就是双列内存模组，实际就是我们常见的装在电脑主板上内存插槽里的内存条，如下图

SO-DIMM

SO-DIMM ,全称是 Small Outline dual in-line memory module，翻译成中文就是小外形双列内存模组。这实际就是装在笔记本电脑主板上内存插槽里的内存条，如下图。由于笔记本空间的限制，SO-DIMM 比普通 DIMM 尺寸更小。

Non-buffer DIMM

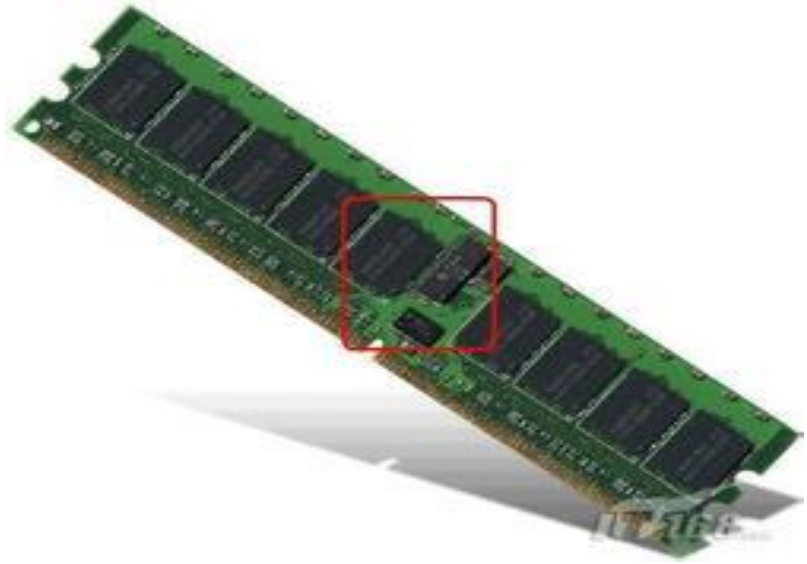
Non-buffer DIMM，其实就是前面一开始介绍的最普通的电脑中常用的 DIMM 内存条。在该内存条上，内存总线是直接由金手指接口连接到内存条上的内存芯片。由于考虑到高速信号完整性的要求，内存总线的信号上连接的内存芯片数量是有限的，也就是说起内存容量大小是有一定限度的。

Buffered DIMM

Buffered DIMM，也称 registered DIMM，是一种在高端产品中比如服务器产品中会用到的内存条。与普通的 DIMM 上内存总线连接所不同的是，这种内存条上的内存总线中的地址信号和控制信号并不是直接连接内存芯片颗粒，而是先连接到内存条上的缓冲器（buffer）。然后经过缓冲器处理后，这些信号从缓冲器输出，再连接到内存条上的内存芯片颗粒。这样做的好处是地址信号和控制信号经过缓冲器处理后，其信号的驱动能力和信号质量大大增强，而且内存条上走线距离较短，故而可以连接到更多的内存芯片颗粒。下图就是一个典型的 Buffered DIMM。图中中间的上下两个长方形的芯片就是缓冲器 buffer，同时该内存条设计单面放置上下两排内存芯片，从而大大增加单根内存条的内存容量。

ECC DIMM

ECC DIMM 就是带错误校验和纠正数据的内存条。该内存条上有一个专门的内存芯片颗粒存放内存数据线上传输的数据的错误校验和纠正，计算机系统可以根据这些信息对一些数据错误进行纠正，从而大大提高系统的运行可靠性和稳定性。下图显示的就是一个 ECC DIMM，图中红色框中的内存芯片就是用来存放错误校验和纠正信息的。这中内存条一般也使用的高端的比如服务器产品的设计中。一般情况下，上节中讲到的 Buffered DIMM 上往往都会带有 ECC 内存芯片，也就是 Buffered DIMM 一般都具有错误校验和纠正功能。下图中的红框中除了 ECC 信息内存颗粒，还有一个长方形的芯片，应该就是前面说的缓冲器 Buffer。当然也有很多 ECC 内存条上并没有这样的缓冲器 – 这样的内存条就只有 ECC 功能，但不支持 Buffer 技术。



Non-ECC DIMM

顾名思义，就是不带 ECC（错误校验和纠正功能）的内存条。这种内存条上的内存芯片都是用来存放内存总线数据总线上传输的数据内容，因而也都连接到内存数据总线。跟 ECC DIMM 内存条比起来，Non-ECC DIMM 每面要少一个内存芯片颗粒，也就是少一棵存放错误校验和纠正信息的内存芯片。

FB-DIMM

FB-DIMM 的全称是 fully buffered DIMM，翻译过来就是全缓存内存模组技术。虽然随着更高速带宽更高的 DDR3/DDR4 技术的出现，FB-DIMM 已推出历史舞台，但为了避免与前面讲过的 buffered DIMM 相混淆，这里还是介绍一下。

FB-DIMM 是 Intel 在 DDR2、DDR3 的基础上发展出来的一种新型内存模组与互联架构，可以搭配现在的 DDR2 及 DDR3 内存芯片。既然是 fully buffered，在这种内存条上也有一个缓冲器 buffer，但这个 buffer 要远比前面说的 buffered DIMM 上的 buffer 功能强大。FB-DIMM 与 buffered DIMM 的主要区别：

- FB-DIMM 与 CPU 内部的内存控制器之间的连接不是传统的并行内存总线，而是速度更快带宽更高的类似 PCI Express 的串行总线；buffered DIMM 则采用传统的并行内存总线技术
- FB-DIMM 上的 buffer 叫 AMB (advanced memory buffer)。这个高大上的内存 buffer，承担实现并行数据流与串行数据流的翻译转换工作和读写控制。其前端管理串行内存数据接口，后端输出传统的并行内存总线，从而连接到 DDR2/DDR3 内存芯片颗粒；buffered DIMM 上的 buffer 功能比较简单，主要是对并行内存总线的地址信号和控制信号进行处理，增强其信号驱动能力。

上图 中显示的就是 FB-DIMM 上的 AMB - 高大上的内存 buffer，可以看到由于其复杂功能，其芯片尺寸也远大于传统的 buffered DIMM 上的 buffer。

下图是一根实际的 FB-DIMM 内存条，可以看到为了保证其散热性能，内存条表面装有长条性散热器。

尽管 FB-DIMM 的前端串行内存总线速度快，带宽高，但其高大上的内存 buffer 功耗大，需要加散热器才能保证其热学性能；价格上 FB-DIMM 也高于 buffered DIMM。随着并行内存总线速度和带宽的持续提升，英特尔处理器/芯片组及各大内存厂商也取消或缩减了 FB-DIMM 内存的生产规模，故 FB-DIMM 内存已经成为历史。

本章内容到这里就全部全部结束了。相信到这里，如果把前面内存芯片信号的功能及相关电路连接都理解了，集成内存设计或内存条电路设计都应当能够读懂。下一章就会针对内存条设计及电路进行全面分析。

后续

微信号：超硬工程师，会继续发布更多使用硬件设计知识和经验，欢迎大家关注。关注后，点击主界面屏幕下方“硬件技术”菜单，就可学习发布的所有硬件技术内容和学习设计经验；点击“资料分享“菜单，可申请相关技术资料。后续内容大概包括：

- 内存及其信号连接电路讲解
- 桥控制器及其信号连接电路讲解
- 电源电路原理及设计讲解

欢迎在微信上扫下面二维码，关注微信号：超硬工程师。