

TFT LCD 液晶显示器的驱动原理

TFT LCD 液晶显示器的驱动原理 (一)

我们针对 TFT LCD 的整体系统面来做介绍，也就是对其驱动原理来做介绍，而其驱动原理仍然因为一些架构上差异的关系，而有所不同。首先我们来介绍由于 Cs(storage capacitor)储存电容架构不同，所形成不同驱动系统架构的原理。

Cs(storage capacitor)储存电容的架构

一般最常见的储存电容架构有两种，分别是 Cs on gate 与 Cs on common 这两种。这两种顾名思义就可以知道，它的主要差别就在于储存电容是利用 gate 走线或是 common 走线来完成的。在上一篇文章中，我曾提到，储存电容主要是为了让充好电的电压，能保持到下一次更新画面的时候之用。所以我们就必须像在 CMOS 的制程之中，利用不同层的走线，来形成平行板电容。而在 TFT LCD 的制程之中，则是利用显示电极与 gate 走线或是 common 走线，所形成的平行板电容，来制作出储存电容 Cs。

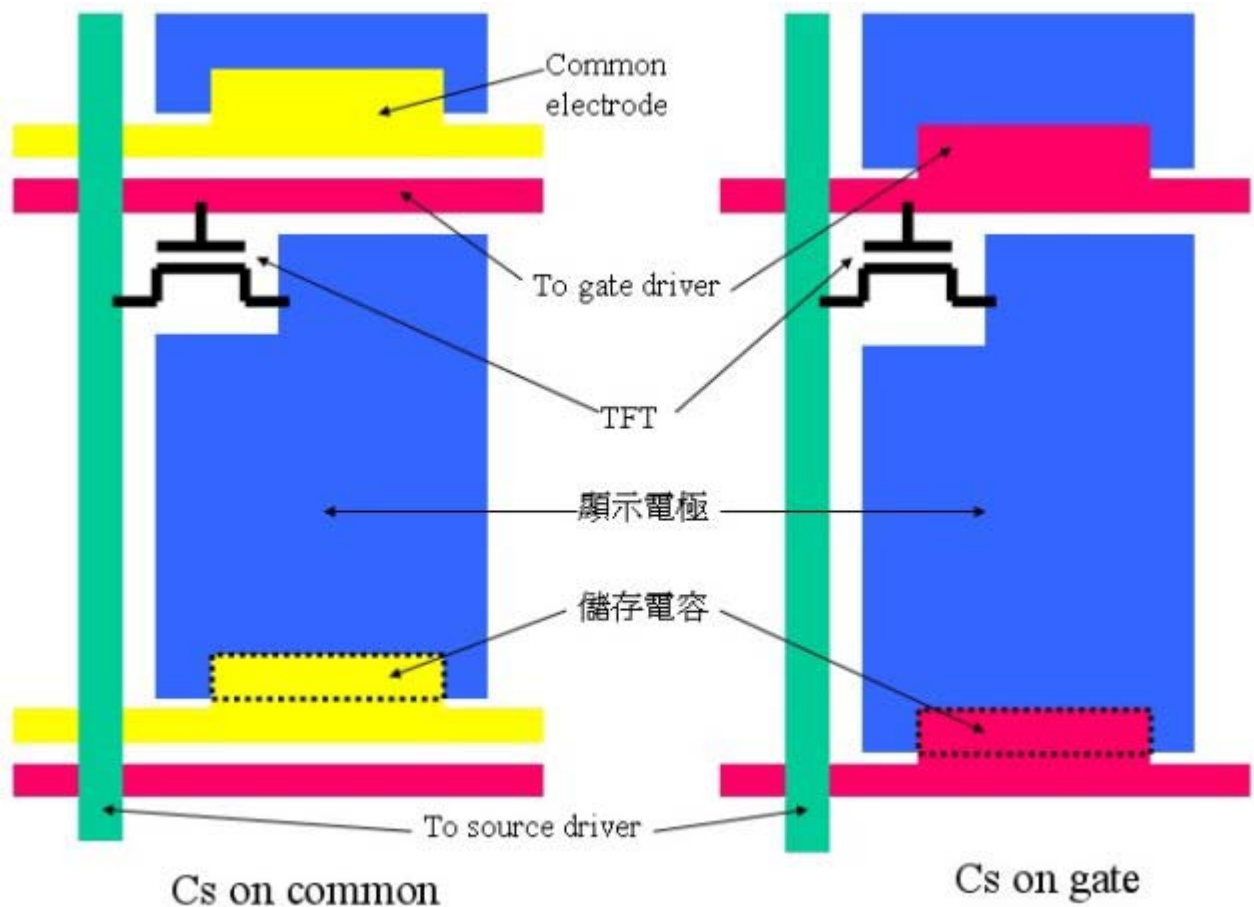


圖1:Cs on gate與Cs on common的架構

图1就是这两种储存电容架构，从图中我们可以很明显的知道，Cs on gate由于不必像Cs on common一样，需要增加一条额外的common走线，所以它的开口率(Aperture ratio)会比较大。而开口率的大小，是影响面板的亮度与设计的重要因素。所以现今面板的设计大多使用Cs on gate的方式。但是由于Cs on gate的方式，它的储存电容是由下一条的gate走线与显示电极之间形成的。(请见图2的Cs on gate与Cs on common的等效电路)而gate走线，顾名思义就是接到每一个TFT的gate端的走线，主要就是作为gate driver送出信号，来打开TFT，好让TFT对显示电极作充放电的动作。所以当下一条gate走线，送出电压要打开下一个TFT时，便会影响到储存电容上储存电压的大小。不过由于下一条gate走线打开到关闭的时间很短，(以1024*768分辨率，60Hz更新频率的面板来说，一条gate走线打开的时间约为20us，而显示画面更新的时间约为16ms，所以相对而言，影响有限。)所以当下一条gate走线关闭，回复到原先的电压，则Cs储存电容的电压，也会随之恢复到正常。这也是为什么，大多数的储存电容设计都是采用Cs on gate的方式的原因。

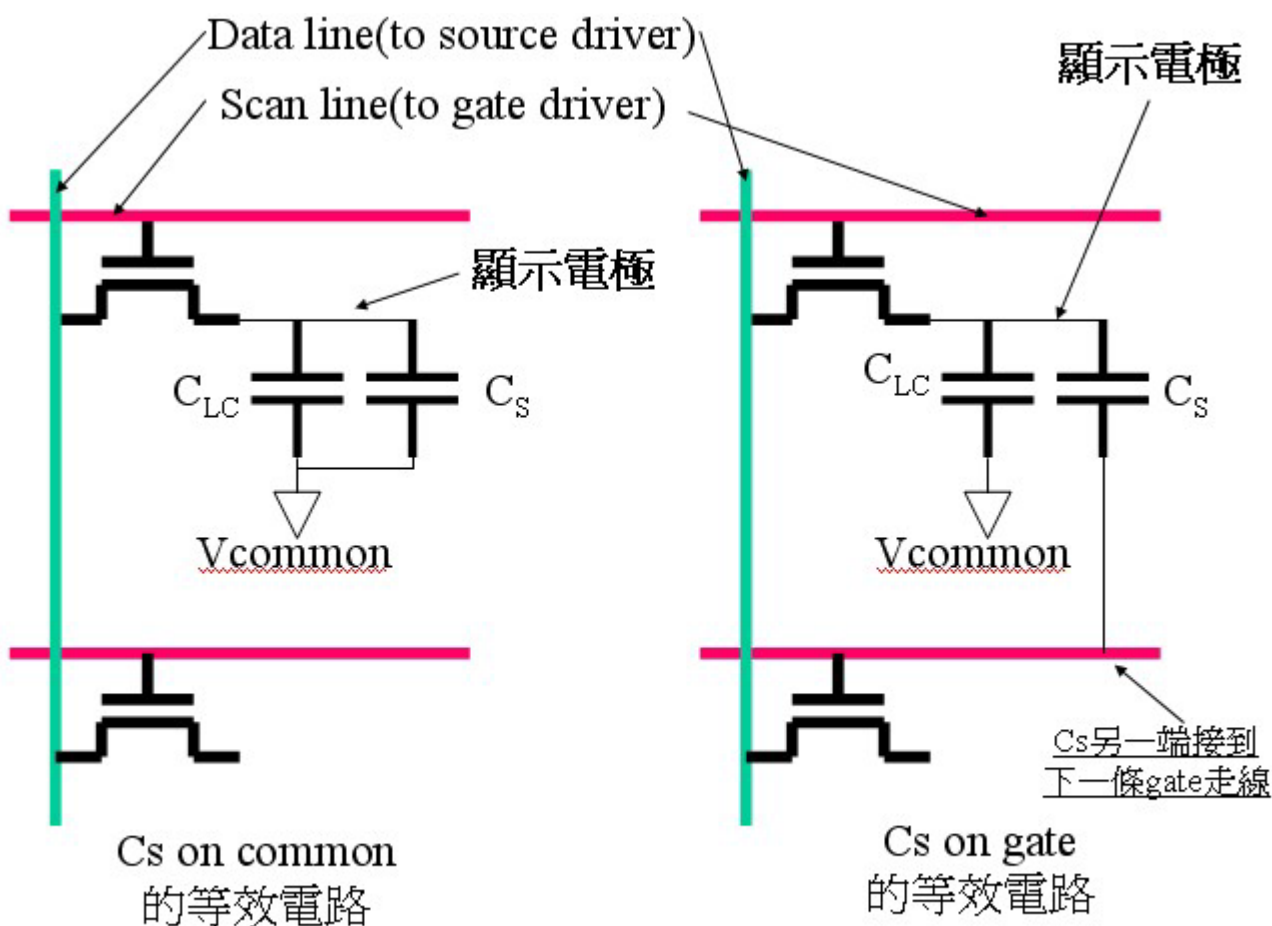


圖2:Cs on gate與Cs on common的等效電路

至于 common 走线，我们在这边也需要顺便介绍一下。从图2中我们可以发现，不管您采用怎样的储存电容架构， C_{LC} 的两端都是分别接到显示电极与common。既然液晶是充满在上下两片玻璃之间，而显示电极与TFT都是位在同一片玻璃上，则common电极很明显

的就是位在另一片玻璃之上。如此一来，由液晶所形成的平行板电容 C_{lc} ，便是由上下两片玻璃的显示电极与 common 电极所形成。而位于 C_s 储存电容上的 common 电极，则是另外利用位于与显示电极同一片玻璃上的走线，这跟 C_{lc} 上的 common 电极是不一样的，只不过它们最后都是接到相同的电压就是了。

整块面板的电路架构

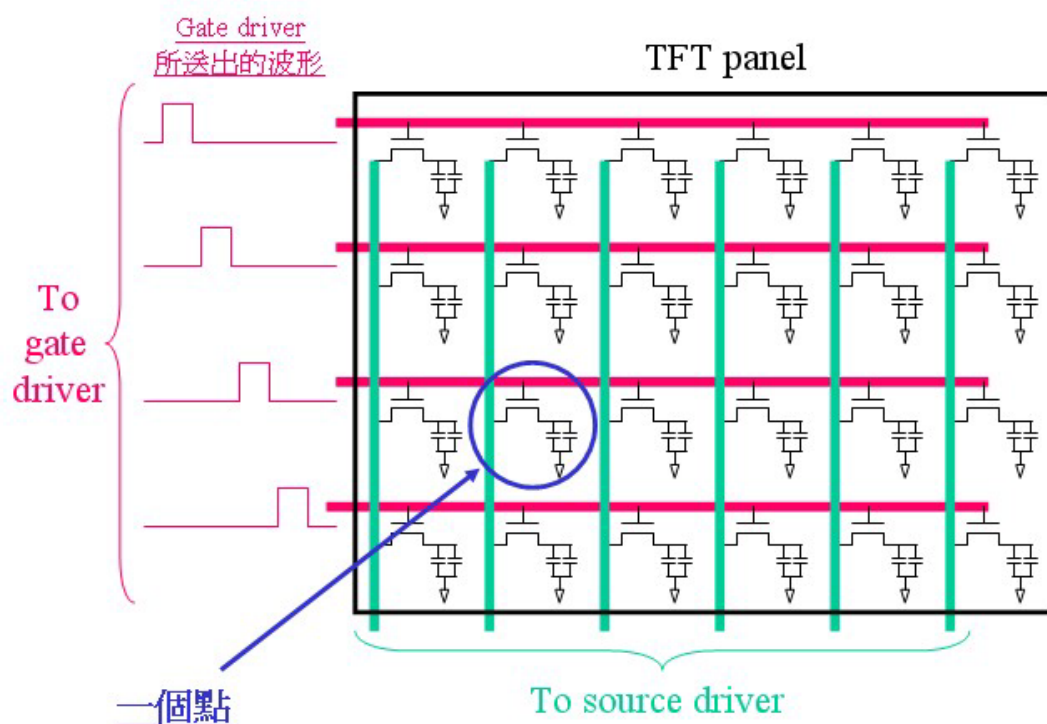


圖3:整塊液晶面板的等效電路

从图 3 中我们可以看到整片面板的等效电路，其中每一个 TFT 与 C_{lc} 跟 C_s 所并联的电容，代表一个显示的点。而一个基本的显示单元 pixel，则需要三个这样显示的点，分别来代表 RGB 三原色。以一个 1024*768 分辨率的 TFT LCD 来说，共需要 1024*768*3 个这样的点组合而成。整片面板的大致结构就是这样，然后再藉由如图 3 中 gate driver 所送出的波形，依序将每一行的 TFT 打开，好让整排的 source driver 同时将一整行的显示点，充电到各自所需的电压，显示不同的灰阶。当这一行充好电时，gate driver 便将电压关闭，然后下一行的 gate driver 便将电压打开，再由相同的一排 source driver 对下一行的显示点进行充放电。如此依序下去，当充好了最后一行的显示点，便又回来从头从第一行再开始充电。以一个 1024*768 SVGA 分辨率的液晶显示器来说，总共会有 768 行的 gate 走线，而 source 走线则共需要 1024*3=3072 条。以一般的液晶显示器多为 60Hz 的更新频率来说，每一个画面的显示时间约为 $1/60=16.67\text{ms}$ 。由于画面的组成为 768 行的 gate 走线，所以分配给每一条 gate 走线的开关时间约为 $16.67\text{ms}/768=21.7\mu\text{s}$ 。所以在图 3 gate driver 送出的波形中，我们就可以看到，这些波形为一个接着一个宽度为 $21.7\mu\text{s}$ 的脉波，依序打开每一行的 TFT。而 source

driver 则在这 21.7us 的时间内，经由 source 走线，将显示电极充放电到所需的电压，好显示出相对应的灰阶。

面板的各种极性变换方式

由于液晶分子还有一种特性,就是不能够一直固定在某一个电压不变，不然时间久了，你即使将电压取消掉，液晶分子会因为特性的破坏，而无法再因应电场的变化来转动，以形成不同的灰阶。所以每隔一段时间，就必须将电压恢复原状，以避免液晶分子的特性遭到破坏。但是如果画面一直不动，也就是说画面一直显示同一个灰阶的时候怎么办？所以液晶显示器内的显示电压就分成了两种极性，一个是正极性，而另一个是负极性。当显示电极的电压高于 common 电极电压时，就称之为正极性。而当显示电极的电压低于 common 电极的电压时，就称之为负极性。不管是正极性或是负极性，都会有一组相同亮度的灰阶。所以当上下两层玻璃的压差绝对值是固定时，不管是显示电极的电压高，或是 common 电极的电压高，所表现出来的灰阶是一模一样的。不过这两种情况下，液晶分子的转向却是完全相反，也就可以避免掉上述当液晶分子转向一直固定在一个方向时，所造成的特性破坏。也就是说，当显示画面一直不动时，我们仍然可以藉由正负极性不停的交替，达到显示画面不动，同时液晶分子不被破坏掉特性的结果。所以当您所看到的液晶显示器画面虽然静止不动，其实里面的电压正在不停的作更换，而其中的液晶分子正不停的一次往这边转，另一次往反方向转呢！

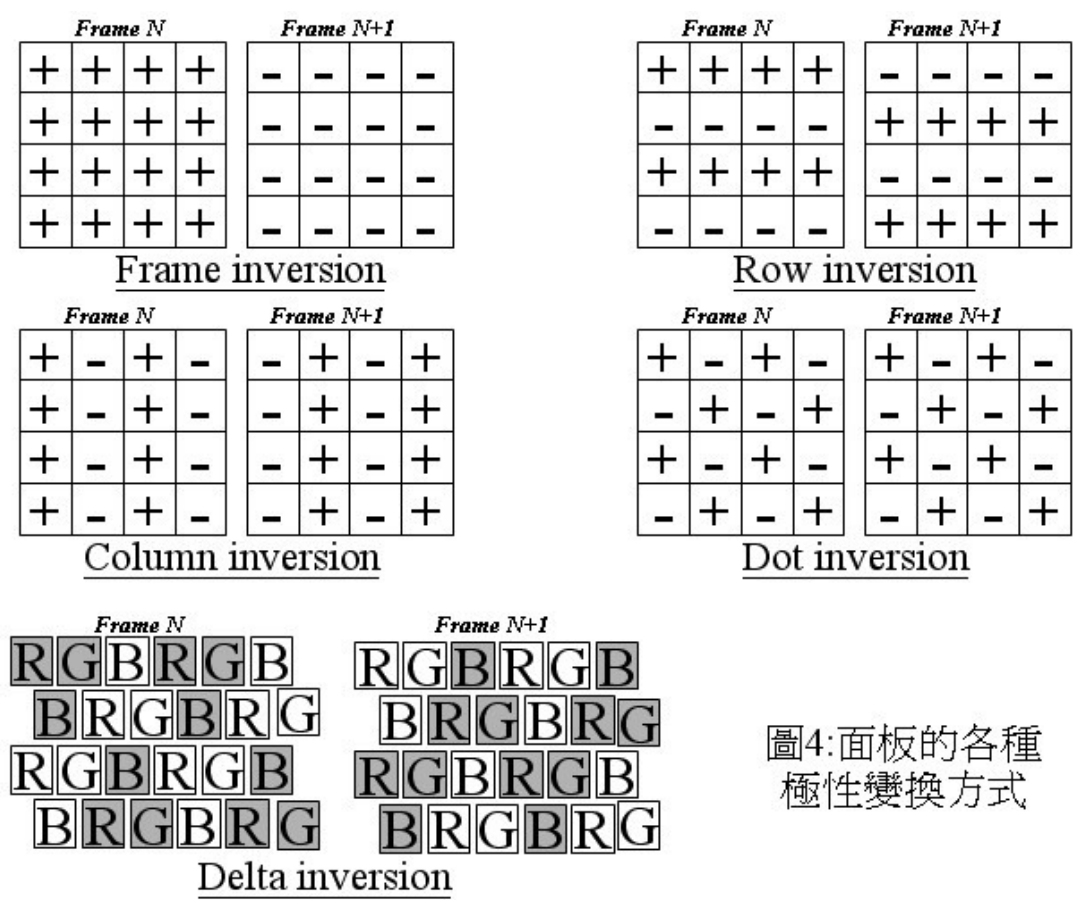


圖4:面板的各種極性變換方式

图 4 就是面板各种不同极性的变换方式，虽然有这么多种的转换方式，它们有一个共通点，都是在下一次更换画面数据的时候来

改变极性。以 60Hz 的更新频率来说，也就是每 16ms，更改一次画面的极性。也就是说，对于同一点而言，它的极性是不停的变换的。而相邻的点是否拥有相同的极性，那可就依照不同的极性转换方式来决定。首先是 frame inversion，它整个画面所有相邻的点，都是拥有相同的极性。而 row inversion 与 column inversion 则各自在相邻的行与列上拥有相同的极性。另外在 dot inversion 上，则是每个点与自己相邻的上下左右四个点，是不一样的极性。最后是 delta inversion，由于它的排列比较不一样，所以它是以 RGB 三个点所形成的 pixel 作为一个基本单位，当以 pixel 为单位时，它就和 dot inversion 很相似了，也就是每个 pixel 与自己上下左右相邻的 pixel，是使用不同的极性来显示的。

Common 电极的驱动方式

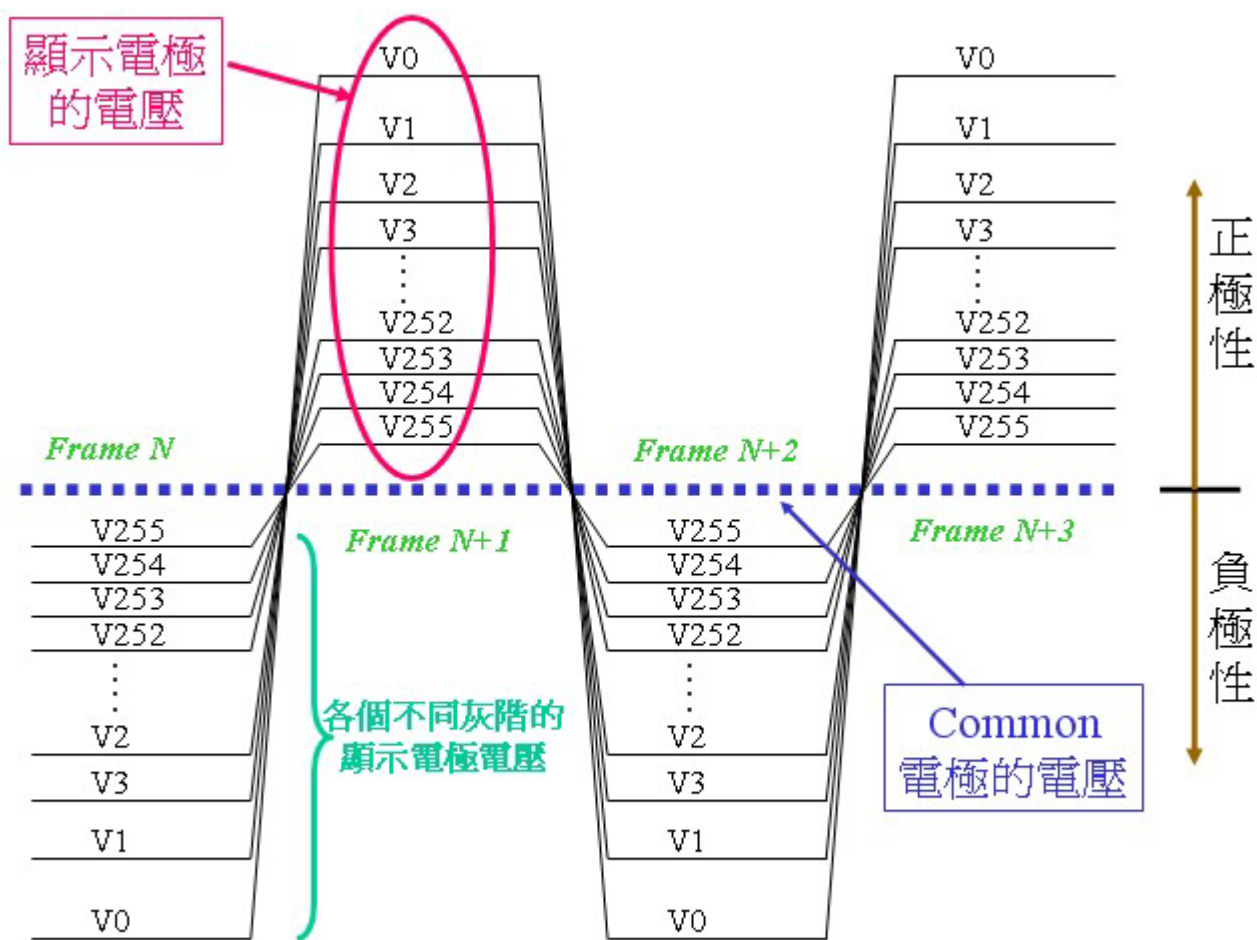


圖5:Common電壓固定不動的驅動方式

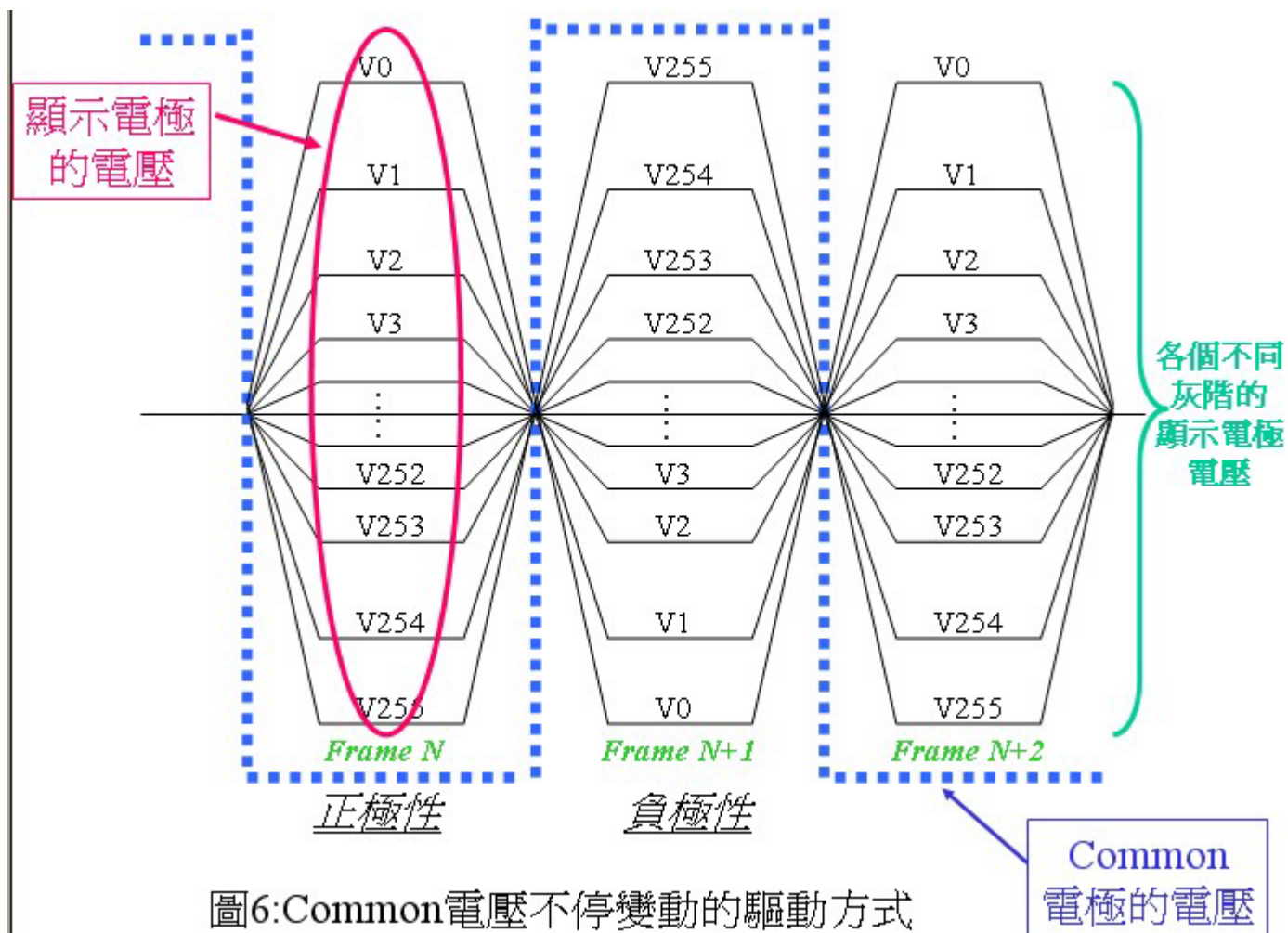


图 5 及图 6 为两种不同的 Common 电极的电压驱动方式，图 5 中 Common 电极的电压是一直固定不动的，而显示电极的电压却是依照其灰阶的不同，不停的上下变动。图 5 中是 256 灰阶的显示电极波形变化，以 V0 这个灰阶而言，如果您要在面板上一直显示 V0 这个灰阶的话，则显示电极的电压就必须一次很高，但是另一次却很低的这种方式来变化。为什么要这么复杂呢？就如同我们前面所提到的原因一样，就是为了让液晶分子不会一直保持在同一个转向，而导致物理特性的永久破坏。因此在不同的 frame 中，以 V0 这个灰阶来说，它的显示电极与 common 电极的压差绝对值是固定的，所以它的灰阶也一直不曾更动。只不过位在 Clc 两端的电压，一次是正的，称之为正极性，而另一次是负的，称之为负极性。而为了达到极性不停变换这个目的，我们也可以让 common 电压不停的变动，同样也可以达到让 Clc 两端的压差绝对值固定不变，而灰阶也不会变化的效果，而这种方法，就是图 6 所显示的波形变化。这个方法只是将 common 电压 一次很大，一次很小的变化。当然啦，它一定要比灰阶中最大的电压还大，而电压小的时候则要比灰阶中最小的电压还要小才行。而各灰阶的电压与图 5 中的一样，仍然要一次大一次小的变化。

这两种不同的 Common 驱动方式影响最大的就是 source driver 的使用。以图 7 中的不同 Common 电压驱动方式的穿透率来说，我们可以看到，当 common 电极的电压是固定不变的时候，显示电极的最高电压，需要到达 common 电极电压的两倍以上。而显示电极电

压的提供，则是来自于 source driver。以图七中 common 电极电压若是固定于 5 伏特的话，则 source driver 所能提供的工作电压

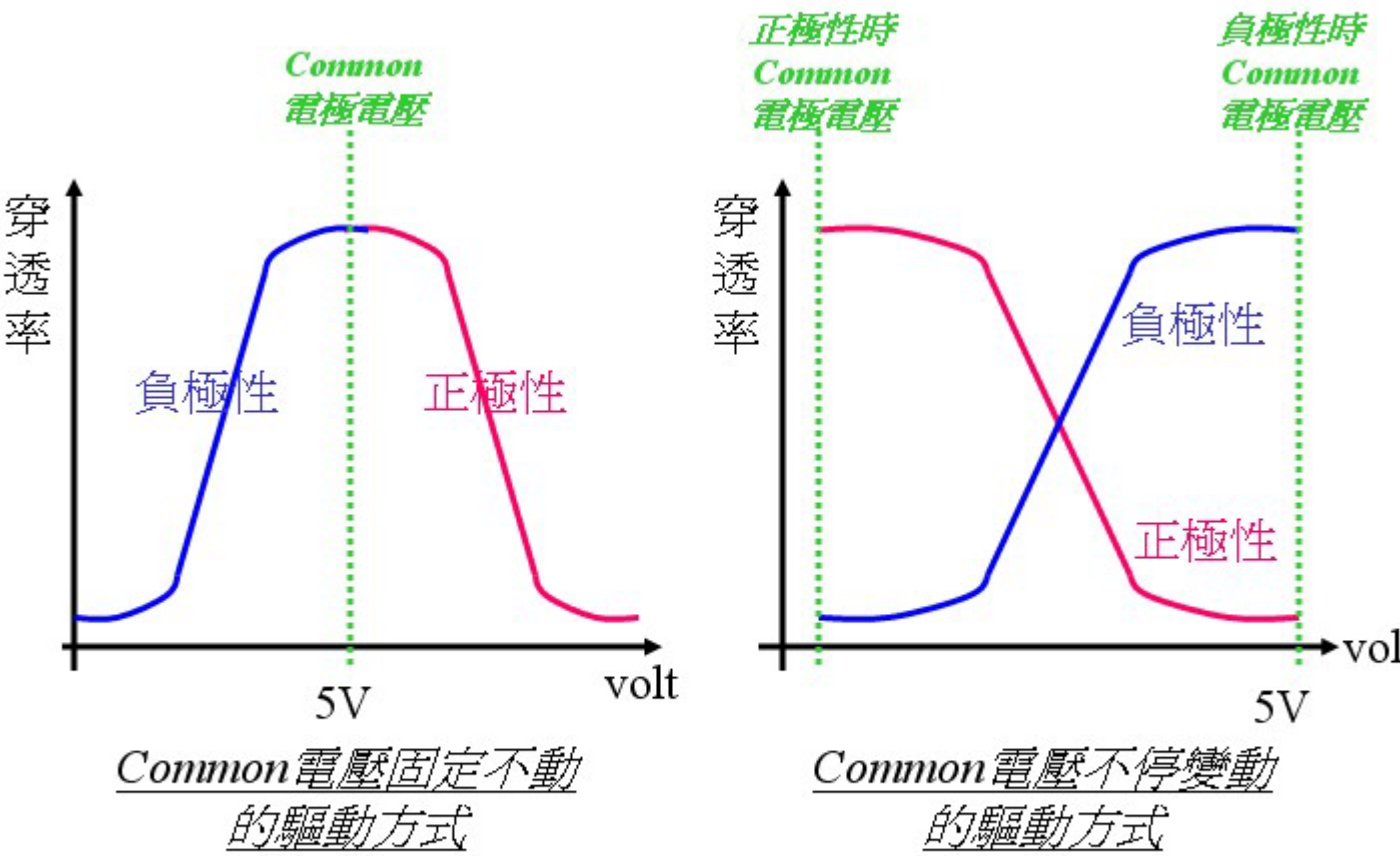


圖7:不同Common電壓驅動方式的穿透率

范围就要到 10 伏特以上。但是如果 common 电极的电压是变动的的话，假使 common 电极电压最大为 5 伏特，则 source driver 的最大工作电压也只要为 5 伏特就可以了。就 source driver 的设计制造来说，需要越高电压的工作范围，制程与电路的复杂度相对会提高，成本也会因此而加高。

面板极性变换与 common 电极驱动方式的选用

并不是所有的面板极性转换方式都可以搭配上上述两种 common 电极的驱动方式。当 common 电极电压固定不变时，可以使用所有的面板极性转换。但是如果 common 电压是变动的的话，则面板极性转换就只能选用 frame inversion 与 row inversion. (请见表 1) 也就是说，如果你想使用 column inversion 或是 dot inversion 的话，你就只能选用 common 电极电压固定不动的驱动方式。为什么呢？之前我们曾经提到 common 电极是位于跟显示电极不同的玻璃上，在实际的制作上时，其实这一整片玻璃都是 common 电极。也就是说，在面板上所有的显示点，它们的 common 电压是全部接在一起的。其次由于 gate driver 的操作方式是将同一行的所有 TFT 打开，

好让 source driver 去充电，而这一行的所有显示点，它的 common 电极都是接在一起的，所以如果你是选用 common 电极电压是可变动的方式的话，是无法在一行 TFT 上，来同时做到显示正极性与负极性的。而 column inversion 与 dot inversion 的极性变换方式，在一行的显示点上，是要求每个相邻的点拥有不同的正负极性的。这也就是为什么 common 电极电压变动的方式仅能适用于 frame inversion 与 row inversion 的缘故。而 common 电极电压固定的方式，就没有这些限制。因为其 common 电压一直固定，只要 source driver 能将电压充到比 common 大就可以得到正极性，比 common 电压小就可以得到负极性，所以 common 电极电压固定的方式，可以适用于各种面板极性的变换方式。

表 1

面板极性变换方式	可使用的 common 电极驱动方式
Frame inversion	固定与变动
Row inversion	固定与变动
Column inversion	只能使用固定的 common 电极电压
Dot inversion	只能使用固定的 common 电极电压

各种面板极性变换的比较

现在常见使用在个人计算机上的液晶显示器，所使用的面板极性变换方式，大部分都是 dot inversion。为什么呢？原因无它，只因为 dot inversion 的显示品质相对于其它的面板极性变换方式，要来的好太多了。表 2 是各种面板极性变换方式的比较表。所谓 Flicker 的现象，就是当你看液晶显示器的画面上时，你会感觉到画面会有闪烁的感觉。它并不是故意让显示画面一亮一灭来做出闪烁的视觉效果，而是因为显示的画面灰阶在每次更新画面时，会有些微的变动，让人眼感受到画面在闪烁。这种情况最容易发生在使用 frame inversion 的极性变换方式，因为 frame inversion 整个画面都是同一极性，当这次画面是正极性时，下次整个画面就都变成了是负极性。假若你是使用 common 电压固定的方式来驱动，而 common 电压又有了一点误差(请见图 8)，

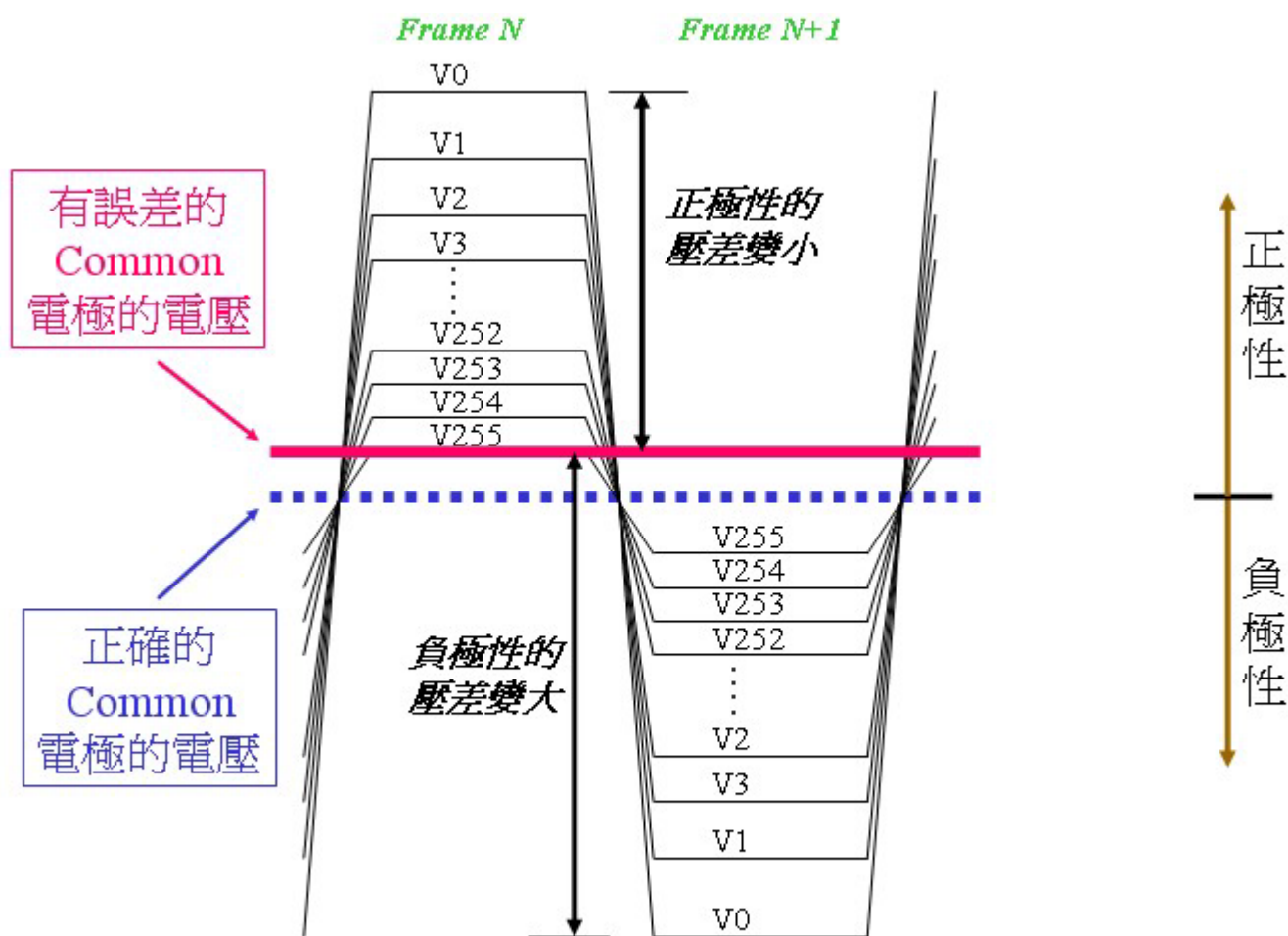


圖8:Flicker的成因

这时候正负极性的同一灰阶电压便会有差别，当然灰阶的感觉也就不一样。在不停切换画面的情况下，由于正负极性画面交替出现，你就会感觉到 Flicker 的存在。而其它面板的极性变换方式，虽然也会有此 flicker 的现象，但由于它不像 frame inversion 是同时整个画面一齐变换极性，只有一行或是一列，甚至于是一个点变化极性而已。以人眼的感觉来说，就会比较不明显。至于 crosstalk 的现象，它指的就是相邻的点之间，要显示的资料会影响到对方，以致于显示的画面会有不正确的状况。虽然 crosstalk 的现象成因有很多种，只要相邻点的极性不一样，便可以减低此一现象的发生。综合这些特性，我们就可以知道，为何大多数人都使用 dot inversion 了。

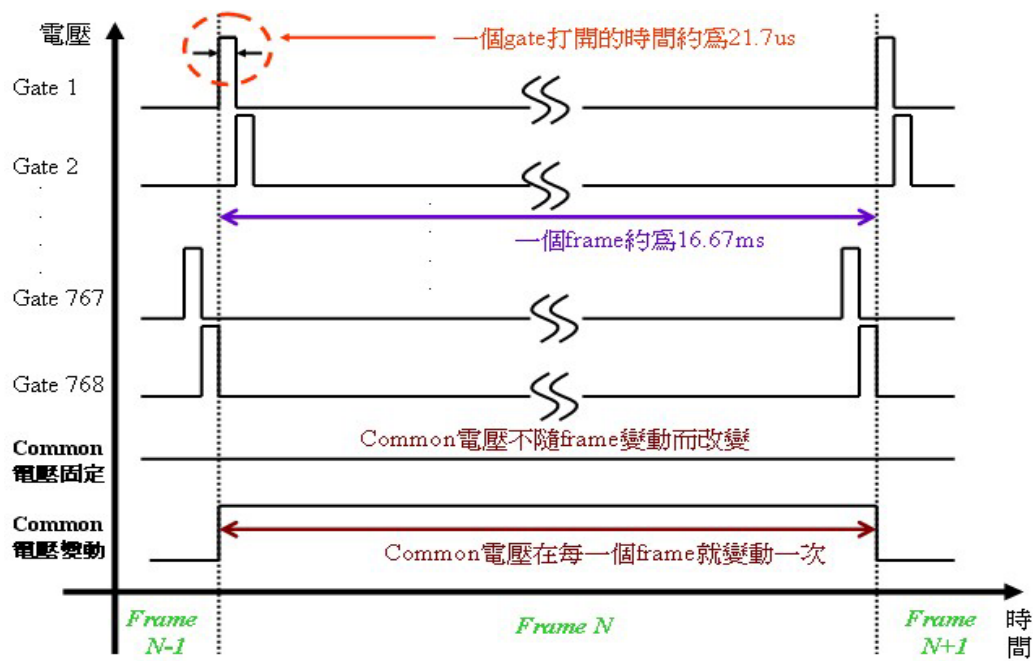
表 2

面板极性变换方式	Flicker 的现象	Crosstalk 的现象
Frame inversion	明显	垂直与水平方向都易发生
Row inversion	不明显	水平方向容易发生
Column inversion	不明显	垂直方向容易发生
Dot inversion	几乎没有	不易发生

面板极性变换方式，对于耗电也有不同的影响。不过它在耗电上需要考量其搭配的 common 电极驱动方式。一般来说 common 电极电压若是固定，其驱动 common 电极的耗电会比较小。但是由于搭配 common 电压固定方式的 source driver 其所需的电压比较高，反而在 source driver 的耗电会比较大。但是如果使用相同的 common 电极驱动方式，在 source driver 的耗电来说，就要考量其输出电压的变动频率与变动电压大小。一般来说，在此种情形下，source driver 的耗电，会有 dot inversion > row inversion > column inversion > frame inversion 的状况。不过现今由于 dot inversion 的 source driver 多是使用 PN 型的 OP，而不是像 row inversion 是使用 rail to rail OP，在 source driver 中 OP 的耗电就会比较小。也就是说由于 source driver 在结构及电路上的改进，虽然先天上它的输出电压变动频率最高也最大(变动电压最大接近 10 伏特，而 row inversion 面板由于多是使用 common 电极电压变动的方式，其 source driver 的变动电压最大只有 5 伏特，耗电上会比较小)，但 dot inversion 面板的整体耗电已经减低很多了。这也就是为什么大多数的液晶显示器都是使用 dot inversion 的方式。

TFT LCD 液晶显示器的驱动原理(二)

上次跟大家介绍液晶显示器的驱动原理中有关储存电容架构，面板极性变换方式，以及 common 电压的驱动方式。这次我们延续上次的内容，继续针对 feed through 电压，以及二阶驱动的原理来做介绍。简单来说 Feed through 电压主要是由于面板上的寄生电容而产生的，而所谓三阶驱动的原理就是为了解决此一问题而发展出来的解决方式，不过我们这次只介绍二阶驱动，至于三阶驱动甚至是四阶驱动则留到下一次再介绍。在介绍 feed through 电压之前，我们先解释驱动系统中 gate driver 所送出波形的 timing 图。



SVGA 分辨率的二阶驱动波形

我们常见的 1024*768 分辨率的屏幕,就是我们通常称之为 SVGA 分辨率的屏幕. 它的组成顾名思义就是以 $1024 \times 768 = 786432$ 个 pixel 来组成一个画面的数据. 以液晶显示器来说,共需要 $1024 \times 768 \times 3$ 个点(乘 3 是因为一个 pixel 需要蓝色,绿色,红色三个点来组成.)来显示一个画面. 通常在面板的规划,把一个平面分成 X-Y 轴来说,在 X 轴上会有 $1024 \times 3 = 3072$ 列. 这 3072 列就由 8 颗 384 输出 channel 的 source driver 来负责推动. 而在 Y 轴上,会有 768 行. 这 768 行,就由 3 颗 256 输出 channel 的 gate driver 来负责驱动. 图 1 就是 SVGA 分辨率的 gate driver 输出波形的 timing 图. 图中 gate 1 ~ 768 分别代表着 768 个 gate driver 的输出. 以 SVGA 的分辨率, 60Hz 的画面更新频率来计算,一个 frame 的周期约为 16.67 ms. 对 gate 1 来说,它的启动时间周期一样为 16.67ms. 而在这 16.67 ms 之间,分别需要让 gate 1 ~ 768 共 768 条输出线,依序打开再关闭. 所以分配到每条线打开的时间仅有 $16.67\text{ms}/768 = 21.7\mu\text{s}$ 而已. 所以每一条 gate driver 打开的时间相对于整个 frame 是很短的,而在这短短的打开时间之内,source driver 再将相对应的显示电极充电到所需的电压.

而所谓的二阶驱动就是指 gate driver 的输出电压仅有两种数值,一为打开电压,一为关闭电压. 而对于 common 电压不变的驱动方式,不管何时何地,电压都是固定不动的. 但是对于 common 电压变动的驱动方式,在每一个 frame 开始的第一条 gate 1 打开之前,就必须把电压改变一次. 为什么要将这些输出电压的 timing 介绍过一次呢?因为我们接下来要讨论的 feed through 电压,它的成因主要是因为面板上其它电压的变化,经由寄生电容或是储存电容,影响到显示电极电压的正确性. 在 LCD 面板上主要的电压变化来源有 3 个,分别是 gate driver 电压变化,source driver 电压变化,以及 common 电压变化. 而这其

中影响最大的就是 gate driver 电压变化 (经由 C_{gd} 或是 C_s), 以及 common 电压变化 (经由 C_{lc} 或是 C_s+C_{lc}).

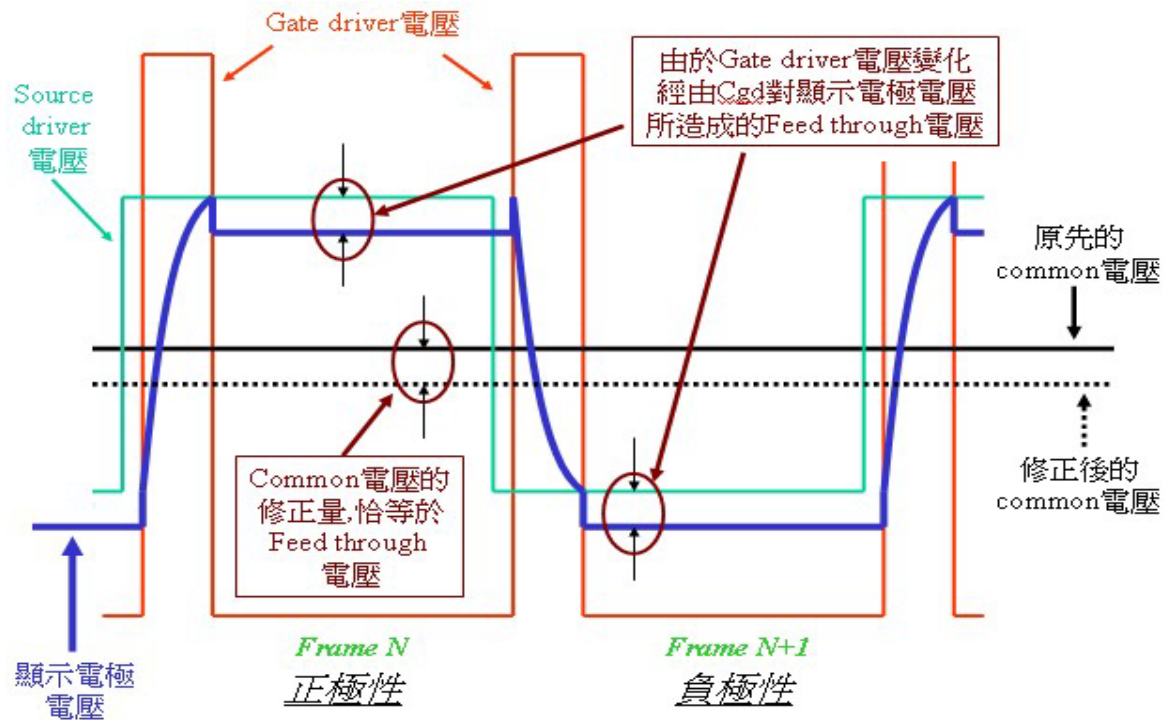


圖2: C_s on common 且 common 电压固定不動的电压波形

C_s on common 架构且 common 电压固定不動的 feed through 电压

我们刚才提到, 造成有 feed through 电压的主因有两个. 而在 common 电压固定不動的架构下, 造成 feed through 电压的主因就只有 gate driver 的电压变化了. 在图 2 中, 就是显示电极电压因为 feed through 电压影响, 而造成电压变化的波形图. 在图中, 请注意 gate driver 打开的时间, 相对于每个 frame 的时间比例是不正确的. 在此我们是为了能仔细解释每个 frame 的动作, 所以将 gate driver 打开的时间画的比较大. 请记住, 正确的 gate driver 打开时间是如同图 1 所示, 需要在 一个 frame 的时间内, 依序将 768 个 gate driver 走线打开的. 所以每个 gate 走线打开的时间, 相对于一个 frame 的时间, 是很短的.

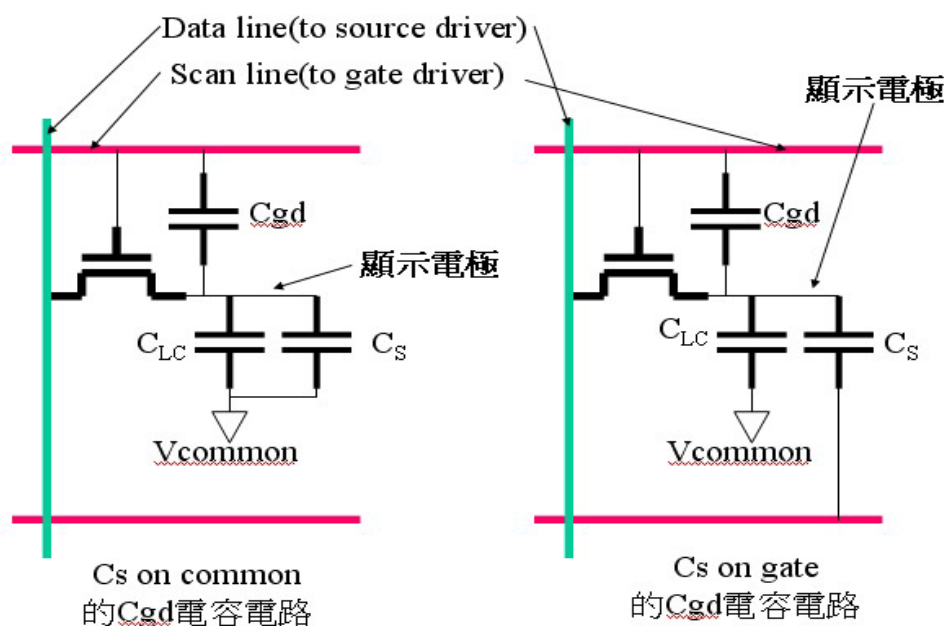
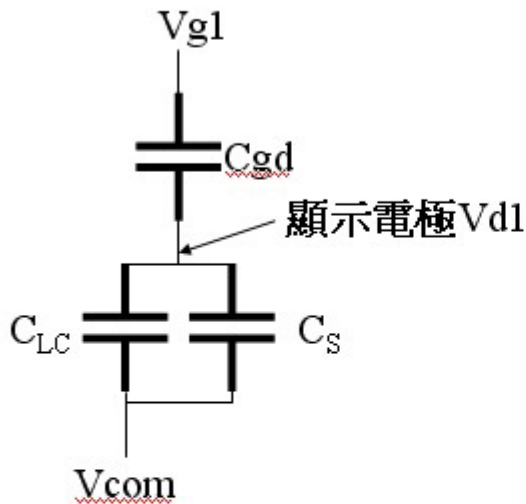


圖3: TFT LCD的Cgd電容電路

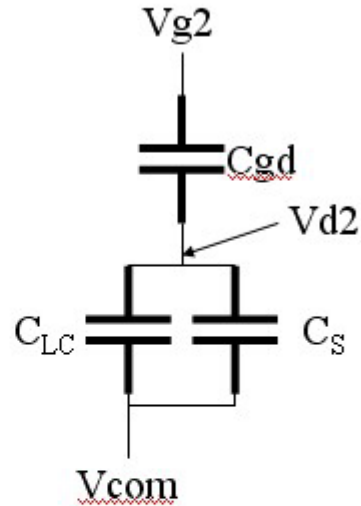
当 gate 走线打开或关闭的那一瞬间, 电压的变化是最激烈的, 大约会有 30~40 伏特, 再经由 Cgd 的寄生电容, 影响到显示电极的电压. 在图 3 中, 我们可以看到 Cgd 寄生电容的存在位置. 其实 Cgd 的发生, 跟一般的 CMOS 电路一样, 是位于 MOS 的 gate 与 drain 端的寄生电容. 但是由于在 TFT LCD 面板上 gate 端是接到 gate driver 输出的走线, 因此一但在 gate driver 输出走在线的电压有了激烈变化, 便会影响到显示电极上的电压. 在图 2 之中, 当 Frame N 的 gate 走线打开时, 会产生一个向上的 feed through 电压到显示电极之上. 不过此时由于 gate 走线打开的缘故, source driver 会对显示电极开始充电, 因此即便一开始的电压不对 (因为 feed through 电压的影响), source driver 仍会将显示电极充电到正确的电压, 影响便不会太大. 但是如果当 gate 走线关闭的时候, 由于 source driver 已经不再对显示电极充电, 所以 gate driver 关闭时的电压压降 (30~40 伏特), 便会经由 Cgd 寄生电容 feed through 到显示电极之上, 造成显示电极电压有一个 feed through 的电压压降, 而影响到灰阶显示的正确性. 而且这个 feed through 电压不像 gate 走线打开时的 feed through 电压一样, 只影响一下子, 由于此时 source driver 已经不再对显示电极充放电, feed through 电压压降会一直影响显示电极的电压, 直到下一次 gate driver 走在线的电压再打开的时后. 所以这个 feed through 电压对于显示画面的灰阶的影响, 人眼是可以明确的感觉到它的存在的. 而在 Frame N+1 的时候, 刚开始当 gate driver 走线打开的那一瞬间, 也会对显示电极产生一个向上的 feed through 电压, 不过这时候由于 gate 已经打开的缘故, source driver 会开始对显示电极充电, 因此这个向上的 feed through 电压影响的时间便不会太长. 但是当 gate 走线再度关闭的时候, 向下的 feed through 电压便会让处在负极性的显示电极电压再往下降, 而且受到影响的负极性显示电压会一直维持到下一次 gate 走线再打开的时候. 所以整体来说, 显示电极上的有效电压, 会比 source driver 的输出电压要低. 而减少的电压大小刚好为 gate 走线电压变化经由 Cgd 的 feed through 电压. 这个电压有多

大呢?

Gate走線打開的時候



Gate走線關閉的時候



$$\begin{aligned} \text{電荷不滅: } & (V_{d1} - V_{g1}) * C_{gd} + (V_{d1} - V_{com}) * (C_{lc} + C_s) \\ & = (V_{d2} - V_{g2}) * C_{gd} + (V_{d2} - V_{com}) * (C_{lc} + C_s) \\ \rightarrow \text{Feed through 電壓} & = V_{d2} - V_{d1} \\ & = (V_{g2} - V_{g1}) * C_{gd} / (C_{gd} + C_{lc} + C_s) \end{aligned}$$

圖4: Gate走線電壓變化時的feed through電壓推導

在图 4 中, 我们以电荷不灭定律, 可以推导出 feed through 电压为 $(V_{g2} - V_{g1}) * C_{gd} / (C_{gd} + C_{lc} + C_s)$. 假设 $C_{gd}=0.05\text{pF}$, 而 $C_{lc}=0.1\text{pF}$, $C_s=0.5\text{pF}$ 且 gate 走线从打开到关闭的电压为 -35 伏特的话. 则 feed through 电压为 $-35 * 0.05 / (0.05 + 0.1 + 0.5) = 2.69$ 伏特. 一般一个灰阶与另一个灰阶的电压差约仅有 30 到 50 mV 而已 (这是以 6 bit 的分辨率而言, 若是 8 bit 分辨率则仅有 3 到 5 mV 而已). 因此 feed through 电压影响灰阶是很严重的. 以 normal white 的偏光板配置来说, 会造成正极性的灰阶会比原先预期的来得更亮, 而负极型的灰阶会比原先预期的来得更暗. 不过恰好 feed through 电压的方向有一致性, 所以我们只要将 common 电压向下调整即可. 从图 2 中我们可以看到, 修正后的 common 电压与原先的 common 电压的压差恰好等于 feed through 电压.

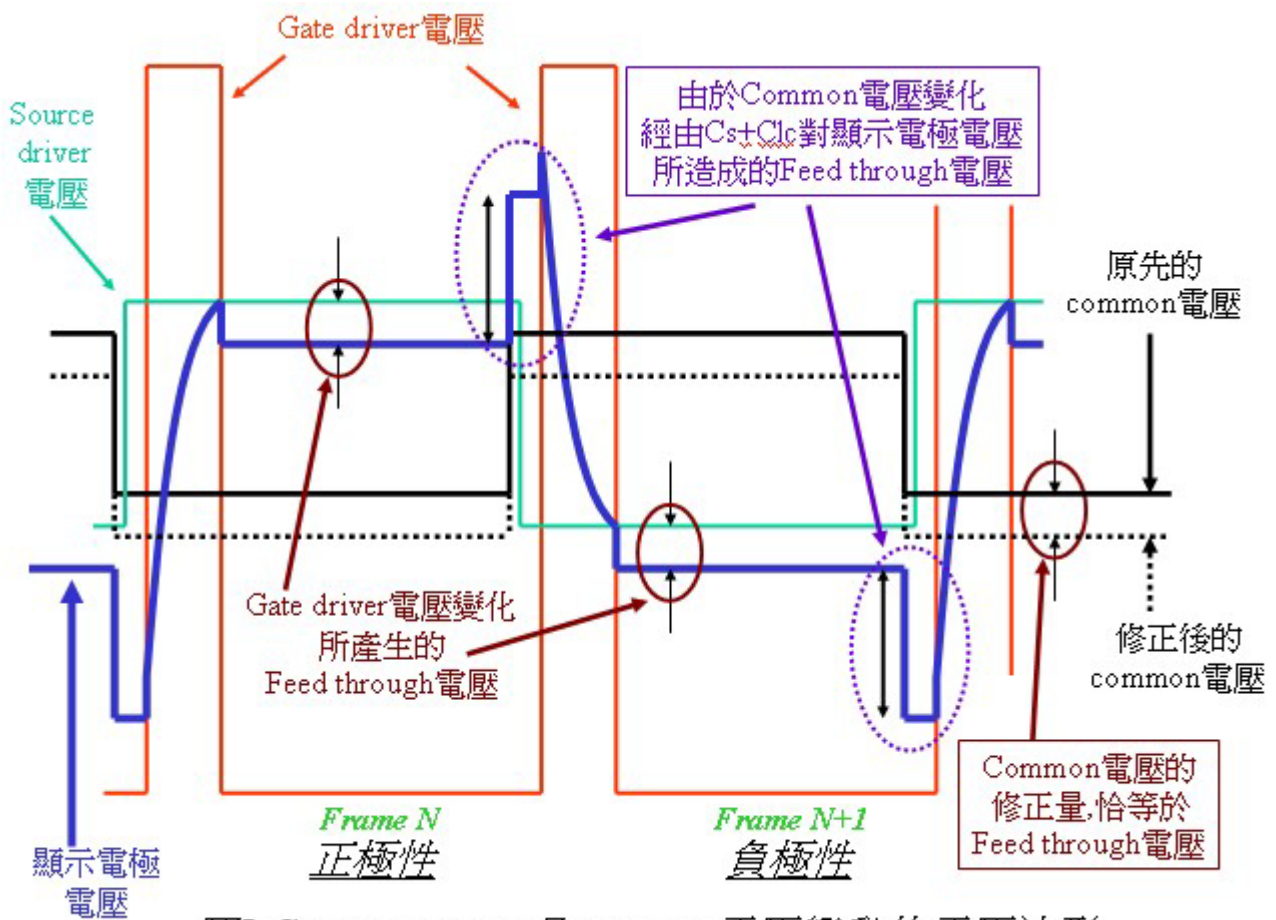


圖5:Cs on common 且common電壓變動的電壓波形

Cs on common 架构且common 电压变动的 feed through 电压

图5为Cs on common 且common 电压变动的电压波形,由于其common 电压是随着每一个frame 而变动的,因此跟common 电压固定的波形比较起来.其产生的feed through 电压来源会再多增加一个,那就是common 电压的变化.这个common 电压的变化,经由Clc+Cs 的电容,便会影响到显示电极的电压.且由于整个LCD 面板上所有显示点的Clc 与Cs 都是接到common 电压,所以一旦common 电压有了变化,受影响的就是整个面板的所有点.跟前面gate 电压变化不一样的是,gate 电压变化影响到的只是一整行的显示点而已.不过Common 电压变化虽然对显示电极的电压有影响,但是对于灰阶的影响却没有像gate 电压变化来的大.怎么说呢?如果我们使用跟前面一样的电容参数值,再套用图6 所推导出来的公式,再假设Common 电压由0 伏特变到5 伏特,则common 电压变化所产生的feed through 电压为 $(5 - 0) * (0.1\text{pF} + 0.5\text{pF}) / (0.05\text{pF} + 0.1\text{pF} + 0.5\text{pF}) = 5 * 0.6 / 0.65 = 4.62$ 伏特.虽然显示电极增加这么多电压,但是common 电极也增加了5 伏特.因此在Clc 两端,也就是液晶的两端,所看到的压差变化,就只有 $4.62 - 5 = 0.38$ 伏特而已.跟之前gate 走线电压变化所产生的feed through 电压2.69 伏特比较起来要小的多了,所以对灰阶的影响也小多了.且由于它所产生的feed through 电压有对称性,不像Gate 走线所产生的feed

through 电压是一律往下, 所以就同一个显示点来说, 在视觉对灰阶的表现影响会比较小. 当然啦, 虽然比较小, 但是由于对整个 LCD 面板的横向的 768 行来说, common 电压变化所发生的时间点, 跟 gate 走线打开的时间间隔并不一致, 所以对整个画面的灰阶影响是不一样的. 这样一来, 就很难做调整以便改进画面品质, 这也是为什么 common 电压变动的驱动方式, 越来越多人使用的缘故.

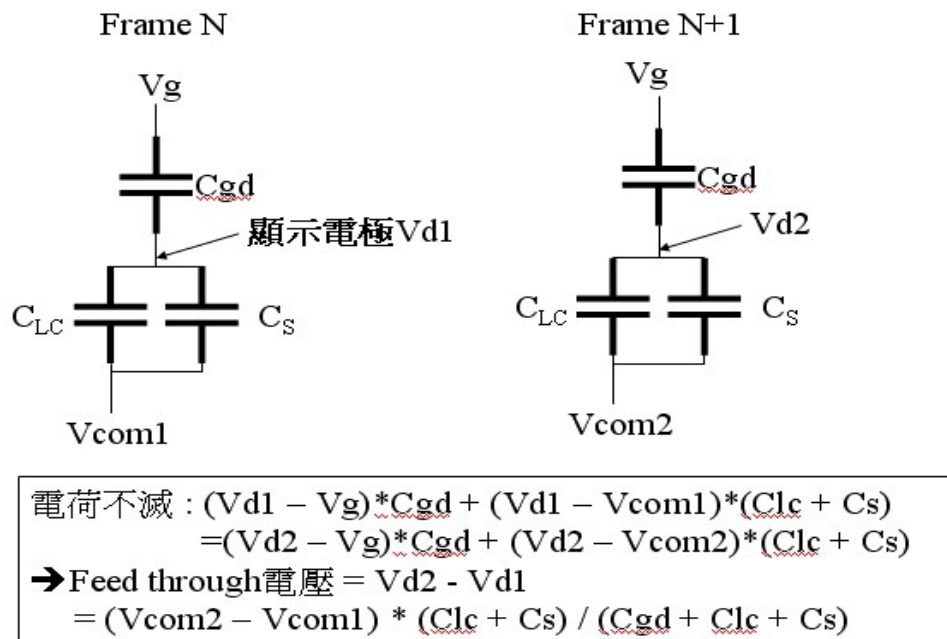


圖6: Common 電壓變化時的 feed through 電壓推導

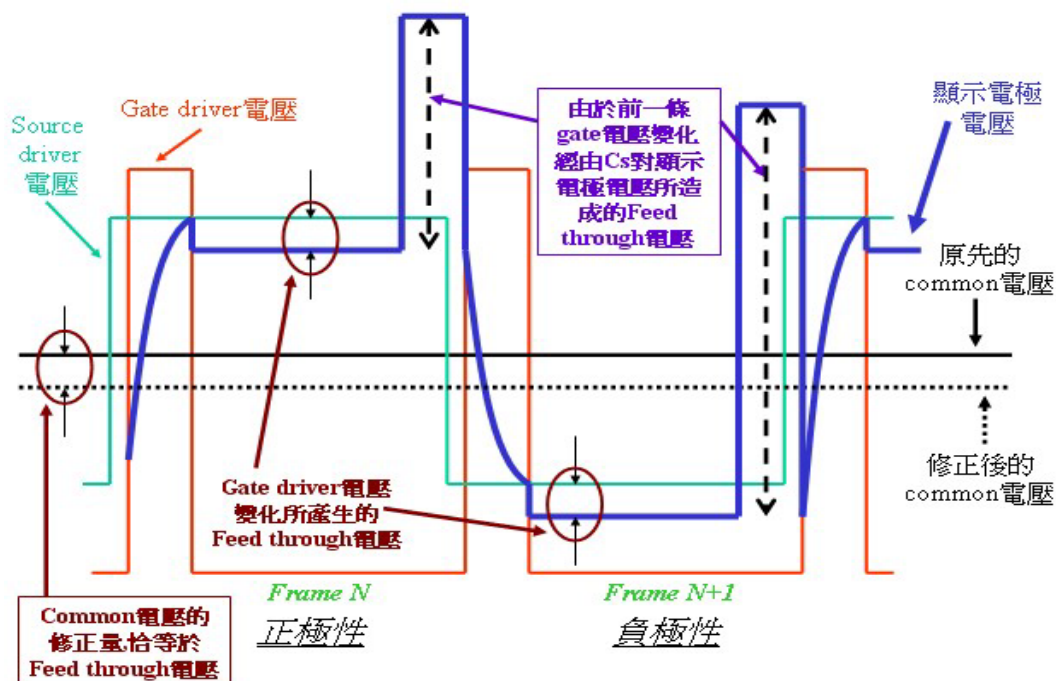


圖7: Cs on gate 且 common 電壓固定不動的電壓波形

Cs on gate 架构且 common 电压固定不动的 feed through 电压

图 7 是 Cs on gate 且 common 电压固定不动的电压波形图. 它并没有 common 电压变化所造成的 feed through 电压, 它只有由于 gate 电压变化所造成的 feed through 电压. 不过它跟 Cs on common 不一样的是, 由 gate 电压变化所造成的 feed through 电压来源有两个地方, 一个是自己这一条 gate 走线打开经由 Cgd 产生的 feed through 电压, 另一个则是上一条 gate 走线打开时, 经由 Cs 所产生的 feed through 电压. 经由 Cgd 的 feed through 电压跟前面所讨论过的状况是一样的, 在这边就不再提了. 但是经由 Cs 的 feed through 电压, 是因为 Cs on gate 的关系, 如图 3 所示. Cs on gate 的架构, 它的储存电容另一端并不是接到 common 电压, 而是接到前一条 gate 走线, 因此在我们这一条 gate 走线打开之前, 也就是前一条 gate 走线打开时, 在前一条 gate 走线的电压变化, 便会经由 Cs 对我们的显示电极造成 feed through 电压. 依照图 8 的公式, 同时套用前面的电容参数与 gate 电压变化值, 我们可得到此一 feed through 电压约为 $35 \times 0.5\text{pF} / (0.5\text{pF} + 0.1\text{pF} + 0.05\text{pF}) = 26.92$ 伏特. 这样的 feed through 电压是很大的, 不过当前一条 gate 走线关闭时, 这个 feed through 电压也会随之消失. 而且前一条 gate 走线从打开到关闭, 以 SVGA 分辨率的屏幕来说, 约只有 $21.7\mu\text{s}$ 的时间而已. 相对于一个 frame 的时间 16.67ms 是很短的. 再者当前一条 gate 走线的 feed through 电压影响显示电极后, 我们这一条的 gate 走线也随之打开, source driver 立刻将显示电极的电压充放电到所要的目标值. 从这种种的结果看来, 前一条 gate 走线的电压变化, 对于我们的显示电极所表现的灰阶, 几乎是没有什么影响的. 因此对于 Cs on gate 且 common 电压固定不动的驱动方式来说, 影响最大的仍然是 gate 走在线电压变化经由 Cgd 产生的 feed through 电压, 而其解决方式跟前面几个一样, 只需将 common 电压往下调整即可.

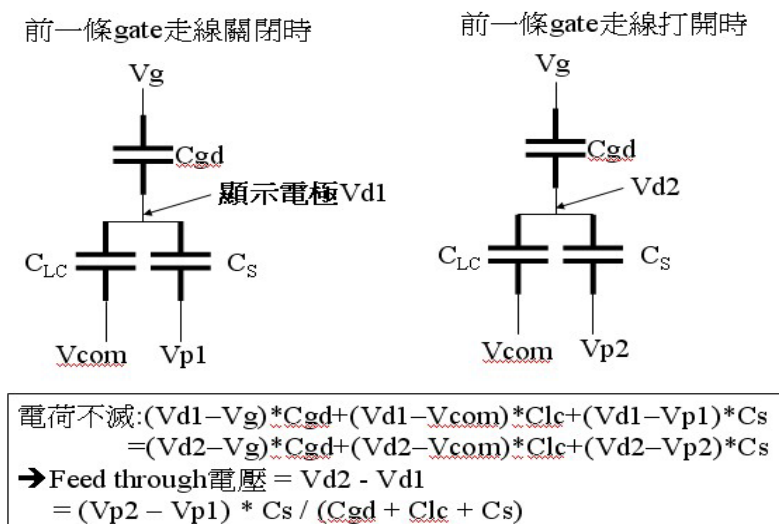


圖8: 前一條gate走線電壓變化時的feed through電壓推導

Cs on gate 架构且 common 电压变动的 feed through 电压

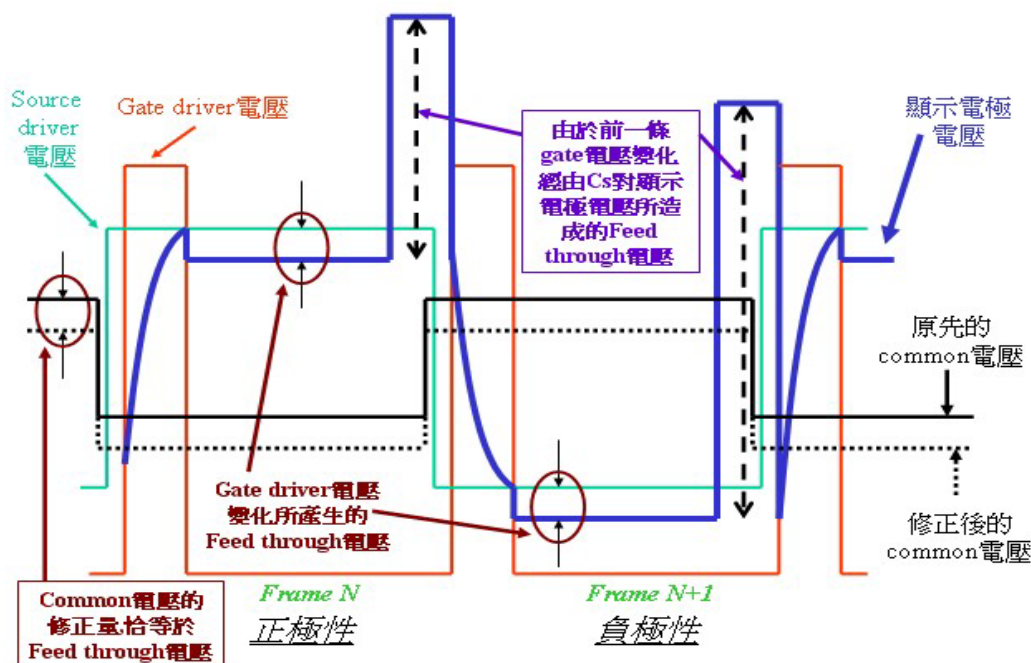


圖9: Cs on gate 且 common 電壓變動的電壓波形

图 9 是 Cs on gate 架构且 common 电压变动的 feed through 电压波形图. 这样子的架构, 刚好有了前面 3 种架构的所有缺点, 那就是 gate 走线经由 Cgd 的 feed through 电压, 和前一条 gate 走线经由 Cs 的 feed through 电压, 以及 Common 电压变化经由 Clc 的 feed through 电压. 可想而知, 在实际的面板设计上几乎是没有人使用这种架构的. 而这 4 种架构中最常用的就是 Cs on gate 架构且 common 电压固定不动的架构. 因为它只需要考虑经由 Cgd 的 feed through 电压, 而 Cs on gate 的架构可得到较大的开口率的缘故.

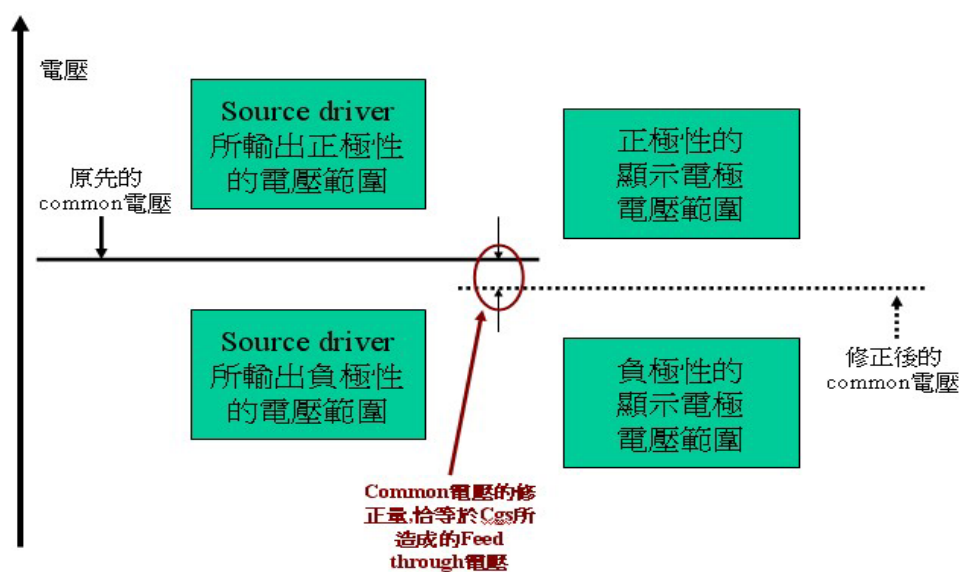


圖10: 二階驅動所需修正電壓的示意圖

二阶驱动(Two level addressing)的效应

TFT LCD 液晶显示器的驱动原理(三)

上次跟大家介绍液晶显示器的二阶驱动原理, 以及因为 feed through 电压所造成的影响. 为了解决这些现象, 于是有了三阶驱动甚至于四阶驱动的设计. 接下来我们先针对三阶驱动的原理作介绍.

三阶驱动的原理(Three level addressing method)

二阶驱动的原理中, 虽然有各种不同的 feed through 电压, 但是影响最大的仍是经由 Cgd 所产生的 feed through 电压. 也因此二阶驱动时需要调整 common 电压, 以改进灰阶品质. 但是由于 Clc 并非是一个固定的参数, 让调整 common 电压以便改进影像品质目的不易达成. 因此便有了三阶驱动的设计, 期望在不变动 common 电压的情形下, 将 feed through 电压给补偿回来.

三阶驱动的基本原理是这样的, 利用经由 Cs 的 feed through 电压, 来补偿经由 Cgd 所产生的 feed through 电压. 也就是因为需要利用 Cs 来补偿, 所以三阶驱动的方法只能使用在面板架构为 Cs on gate 的方式. 图 1 就是三阶驱动 gate driver 电压的波形, 从这个三阶驱动的波形中我们可以知道, 三阶驱动波形跟二阶驱动不一样的是, 它的 gate driver 驱动波形之中, 会有三种不一样的电压. 当 gate driver 关闭时, 会将电压拉到最低的电压, 等到下一条的 gate driver 走线也关闭后, 再将电压拉回. 而这个拉回的电压, 就是为了去补偿下一条线的 feed through 电压. 也就是说, 每一条 gate driver 走线关闭时, 经由 Cgd 所产生的 feed through 电压, 是由上一条走线将电压拉回时, 经由 Cs 所产生的 feed through 电压来补偿的. 既然是经由拉回的电压来补偿, 那拉回电压的大小要如何计算呢? 上次我们有提到 feed through 电压的计算方式, 我们可以依照上次的公式来计算所需的电压:

经 Cgd 的 Feed through 电压 = $(V_{g_high} - V_{g_low}) * C_{gd} / (C_{gd} + C_{lc} + C_s)$; V_{g_high} 与 V_{g_low} 分别为 gate driver 走线打开与关闭的电压.

经 Cs 的 Feed through 电压 = $(V_{p2} - V_{p1}) * C_s / (C_{gd} + C_{lc} + C_s)$; V_{p2} 与 V_{p1} 分别为上一条 ga

te 走线拉回前与拉回后的电压.

如果需要两者互相抵消, 则经 C_{gd} 的 Feed through 电压需要等于经 C_s 的 Feed through 电压. 所以需拉回的电压为 $V_e = V_{p2} - V_{p1} = (V_{g_high} - V_{g_low}) * C_{gd} / C_s$, 而从图 1 中我们知道 $V_{g_high} - V_{g_low} = V_g + V_e$, 所以需拉回的电压 $V_e = (V_g + V_e) * C_{gd} / C_s$, 也就是 $V_e = V_g * C_{gd} / [C_s - C_{gd}]$.

从上述的公式推导中, 我们发现虽然 C_{lc} 会影响 feed through 电压的大小, 但是藉由三阶驱动的方式, C_{lc} 的影响就不见了. 因此当我们在面板制程与 gate driver 的打开电压确定之后, 就可以精确的计算出所需要的拉回电压了.

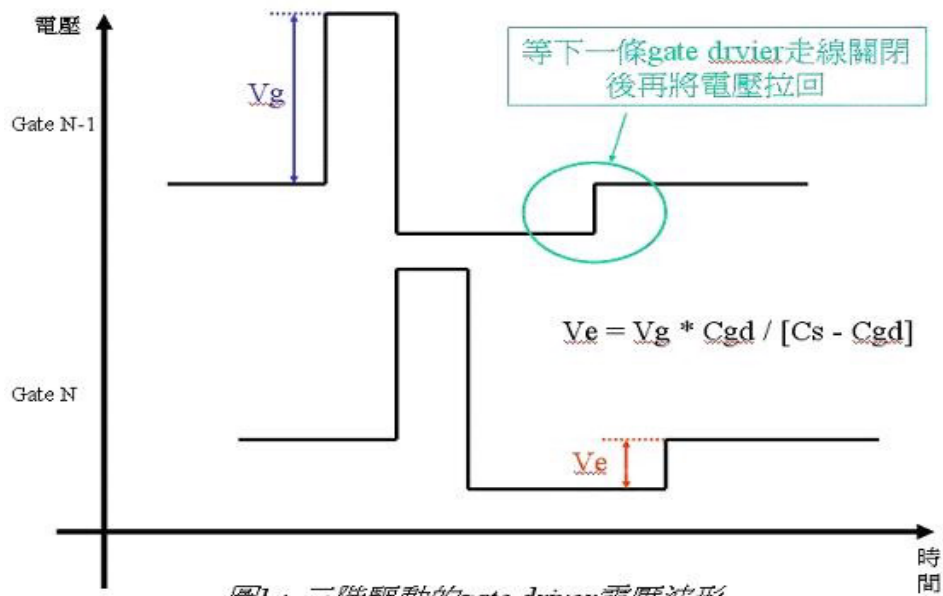


圖1：三階驅動的gate driver電壓波形

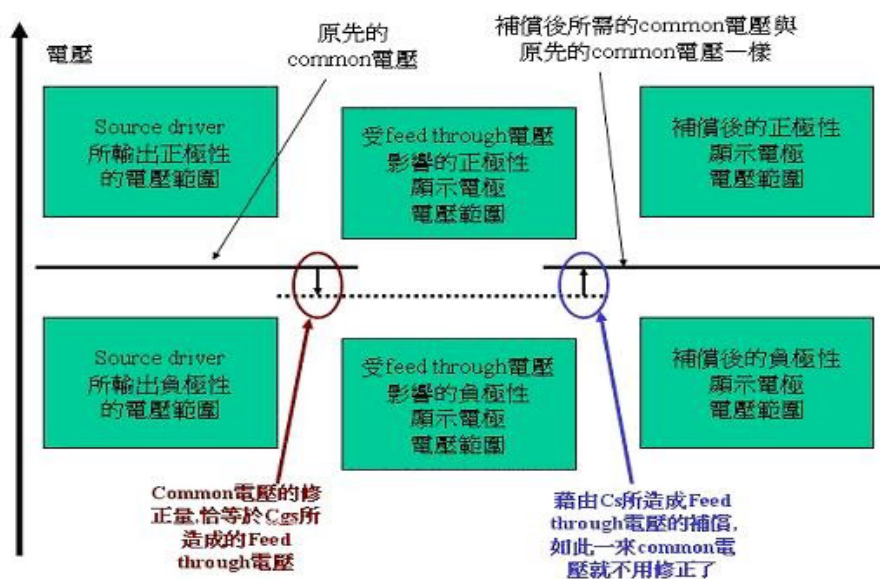


圖2: 三階驅動的電壓分布示意圖

图 2 是三阶驱动的电圧分布示意图。我们可以看到最左边的是由 source driver 所输出的电压分布，这是显示电极所充电电压的最原始状况。而中间的电圧分布，就是显示电极受到经由 C_{gd} 的 feed through 电压影响的变化。一般二阶驱动就是只有到这里，所以需要修正 common 电压的大小，以便以少灰阶的失真程度。而三阶驱动藉由 C_s 的 feed through 电压影响的情形，则可以由最右边的电压分布来看出。在这时候，只要拿捏好拉回电压 V_e 的大小，便可以将原本受到经由 C_{gd} 的 feed through 电压影响的电压分布，补偿到跟最左边的电压分布一样，如此一来就不必再去修正 common 电压的大小了。

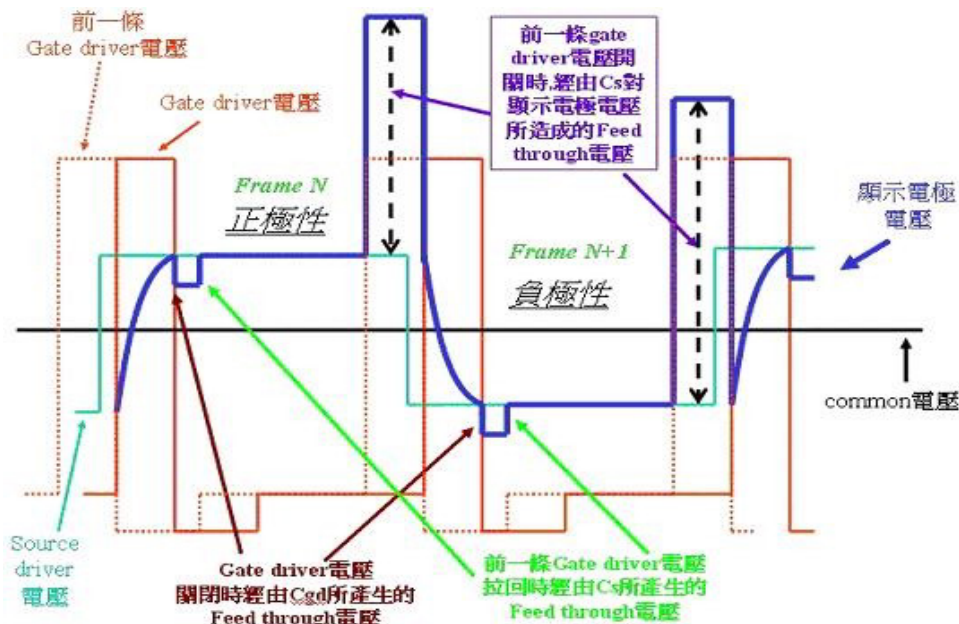
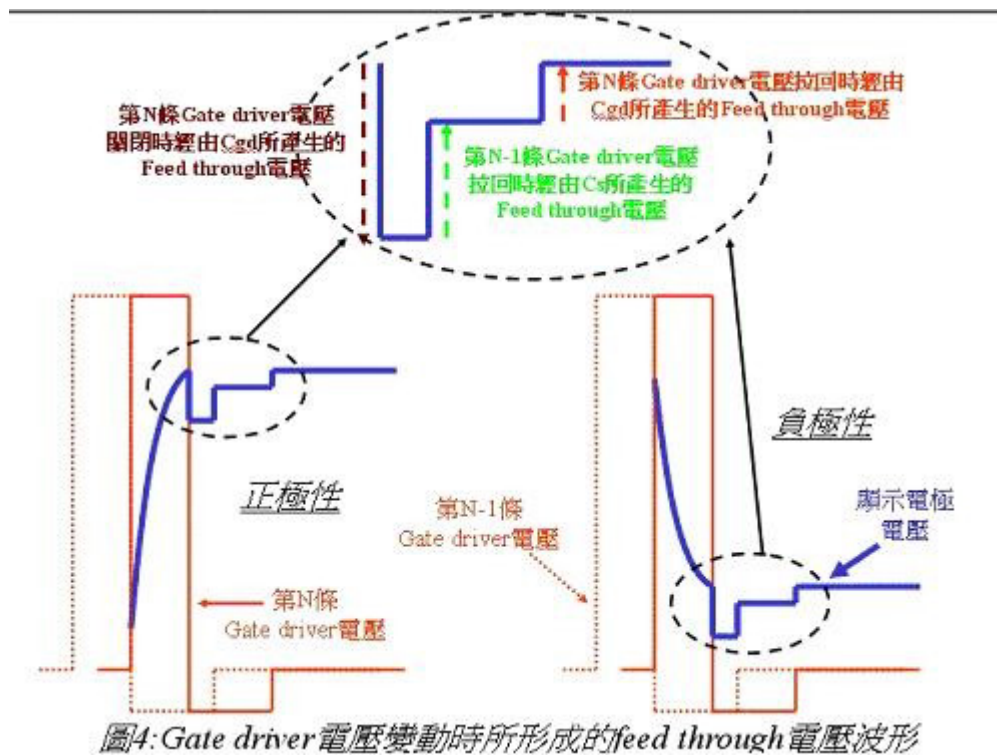


圖3: Cs on gate 且 common 電壓固定不動的三階驅動電壓波形

图3是三阶驱动的电壓波形图。正如先前所说的，由于三阶驱动需要利用前一条的 gate driver 走线来补偿，所以只能使用于 Cs on gate 的架构。而且由于有电压补偿的关系，common 电压就不必再做修正了。在图3中，属于 gate driver 电压有两种，一个是前一条 gate driver 的电壓波形，用虚线来表示，而用实线表示的是属于打开我们要讨论的显示电极电压波形的 gate driver 走线电压。从此图形我们可以知道，实线的 gate driver 走线关闭时，会经由 Cgd 产生一个 feed through 电压，而这个向下的电压偏移量，在前一条 gate driver 走线的拉回电压经 Cs 所产生的 feed through 电压影响后，便可以令显示电极恢复到原先的电压准位。而前一条 gate driver 走线经由 Cs 的 Feed through 电压还有另一种状况，那就是在前一条 gate driver 走线打开时所产生的 feed through 电压，这个电压值虽然很大，不过由于其影响的时间，相对于整个 frame 来说，相当的短，因此对显示画面并不会有多大的影响。

图4是使用三阶驱动针对 gate driver 走线电压变动所形成的 feed through 电压更仔细的显示电极电压波形图。跟图三不一样的，这个图形有考虑到当 gate driver 走线电压拉回时经由 Cgd 所造成的 feed through 电压。原本拉回电压是为了补偿下一条 gate driver 走在线的显示电极，但是它的副作用就是也会对 gate driver 走线所在位置的显示电极产生影响。所以拉回电压的设计考量，并不是一次将所有电压补偿回来，而是使用两次的 feed through 电压补偿。一次是上一条 gate driver 走线经由 Cs 的 feed through 电压来补偿，一次则藉由显示电极所在位置的 gate driver 走线，它的拉回电压经由 Cgd 的 feed through 电压来补偿。

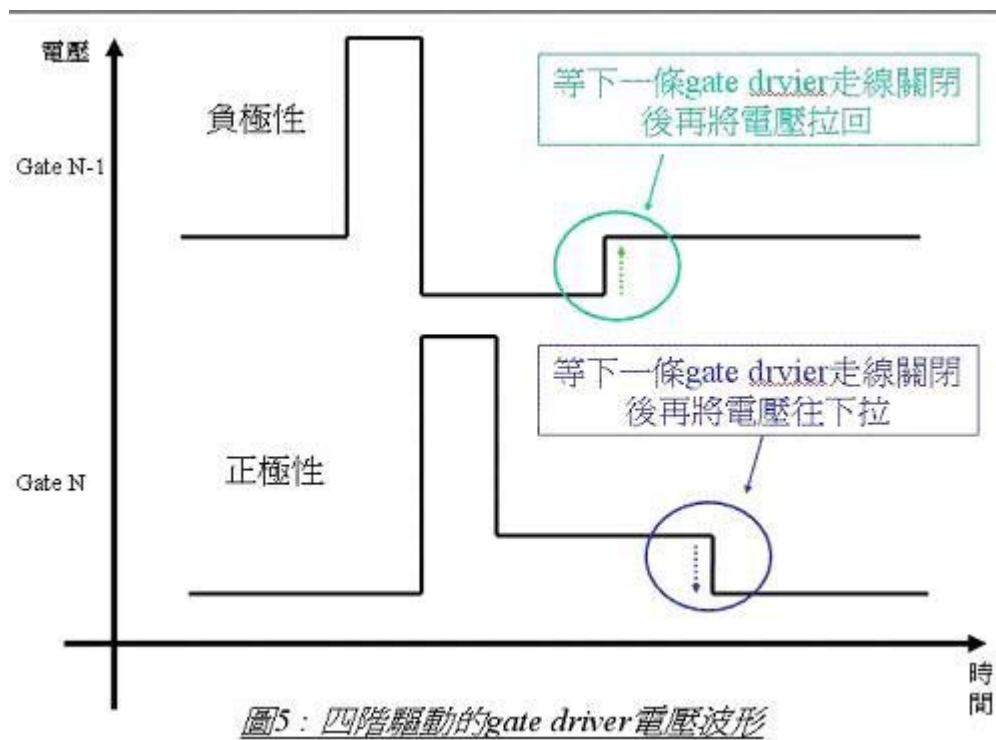


总的来说，使用三阶驱动的方式比起二阶驱动的方式来说，可以不用调整 common 电压就可以克服 feed through 电压的影响。而且也可以避免由于 C_{lc} 的非线性关系所造成的灰阶问题。不过跟底下要介绍的四阶驱动比较起来，它仍然需要使用较高输出电压的 source driver。接下来要介绍的四阶驱动，它在 common 电压固定不变的状况下，并不需要使用高电压输出的 source driver, 就可以达到分别出正负极性电压的结果了。

四阶驱动的原理 (Four level addressing method)

图 5 是四阶驱动 gate driver 走线的电压基本波形。我们可以看到负责正极性与负极性的 gate driver 走线电压是不一样的。负责负极性的 gate driver 走线电压在电压关闭时，会往下拉到一个比一般关闭时的电压更低的准位，等到下一条走线的电压关闭后，再将电压拉回到一般关闭电压的准位。而负责正极性的 gate driver 走线电压则是在电压关闭时，电压并没有一口气拉到一般关闭的电压准位，而是等到下一条 gate driver 走线关闭后，再将电压下拉到一般关闭的电压准位。而这两种极性的电压准位总共有：打开的电压，关闭的电压，比关闭电压高的准位，以及比关闭电压更低的电压，总共四种。这是为什么叫做四阶驱动的原因。从图五来看，我们会发现，同样一条 gate driver 走在线的显示电极，都必须属于同一种显示的极性，不是正极性，就是负极性。因此采用四阶驱动就只能使用 line inversion 的显示方式。不过这样一来，跟使用 dot inversion 驱动方式的面板来说，显示画面的品质

变会变的更差, flicker 与 cross talk 的效应会更明显. 这也是为什么四阶驱动很少有人使用的缘故, 虽然它可以使用驱动电压较低的 source driver, 但是它的 gate driver 复杂度升高, 而且画面品质下降, (当然啦, 想要让四阶驱动的面板使用 dot inversion 并不是不可以, 只是需要更改面板上的 TFT 薄膜晶体管的配置方式, 以及加大显示控制器内的内存大小, 来同时储存两条 gate driver 走在线的所有显示电极的数据, 整个硬件的复杂度会更高, 成本又会加大.) 比较起来倒不如使用 line inversion 且 common 电压变动的面板极性显示方式.



四阶驱动原理简单的来说, 是利用前一个 gate driver 走线经由 Cs 的 feed through 电压, 在正极性时将显示电极的电压提升到很高的电压, 而在负极性时 将显示电极的电压, 下拉到很低的电压, 以便将显示电极的电压分别出给正极性或是负极性的电压位准之用. 如此一来, source driver 的驱动电压范围虽然不大, 但是却可以同时给正极性以及负极性的显示电极电压来用. 图 6 是四阶驱动电压分布示意图, 图中最左边的是 source driver 输出电压的范围. 不管是正极性的画面, 或是负极性的画面, 都是使用相同的输出电压范围. 因此使用于四阶驱动的 source driver, 其输出电压范围比起一般的 source driver 要小的多. 而图 6 中间则是受到 gate driver 走线关闭时, 经由 Cgd 的 feed through 电压影响的显示电极电压范围. 而图 6 右边则是最后分别出正负极性的显示电压范围. 从图中我们可以知道, 因为受到经过 Cgd 的 feed through 电压影响, 若是要将正负极性的电压范围分开的话, 对于正极性的电压范围, 往上提升的电压会比较大, 而其往上提升的电压, 是由上一条 gate drive 走线电

压往上拉经由 C_s 的 feed through 电压来形成。因为其所需的电压比较大，所以上一条 gate driver 走线的拉回电压 也会比较大。而对于负极性的显示电压范围的形成，也是利用上一条 gate driver 走线的电压变化来完成。跟正极性的显示电极电压不一样的是，它需要的是下拉的 feed through 电压，以便形成负的显示电极电压范围。它所需要的下拉电压 跟正极性的上拉电压比较起来会比较小。不过对于调整后正负极性的显示电压范围来说，它们相对于 common 电压的距离要一样，这样对于同一个灰阶的正负极性电压，显示出来的画面才会一致。从整个图 6 来说，我们可以发现，对于 source driver 的输出电压，如同前面所说的，并不需要正负两种不同极性的显示范围。只要利用上一条 gate driver 走线的电压变化来帮助形成正负极性的两种电压范围即可。

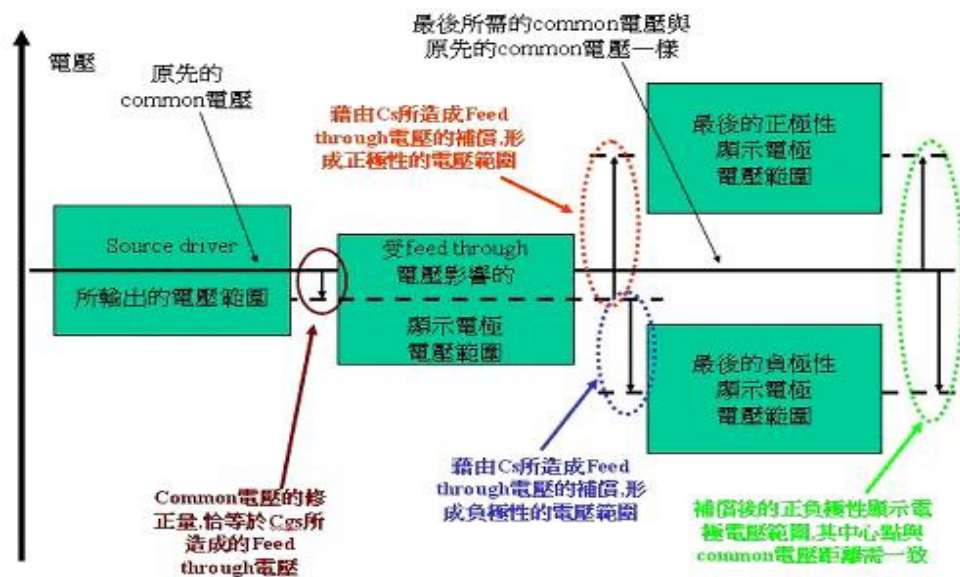


圖6: 四階驅動的電壓分布示意圖

至于在显示电极上的电压变化波形，我们则可以利用图 7 跟图 8 来解释其电压变化的原理。图 7 是负极性显示电极电压的波形变化，从图中我们可以知道 显示电极电压从 source driver 充好电后，会再经过三次的电压变化。首先是本身 gate driver 走线电压关闭时，经由 C_{gd} 所产生的 feed through 电压影响，由于电压关闭的关系，会把显示电极的电压往下拉。其次是一条 gate driver 走线下拉时，经过 C_s 的 feed through 电压。这个电压的影响很重要，因为它将电压调整成负极性电压的主要成分，必须能够将整体的电压调整到所需要的准位。最后是本身的 gate driver 走线电压拉回时，经由 C_{gd} 的 feed through 电压的影响。由于拉回电压的幅度比较小，所以整体的影响也比较少。而图 8 是正极性显示电极电压的波形变化，跟负极性的电压变化一样，它也有三阶段的电压变化。首先是本身 gate driver 走线关闭时经由 C_{gd} 的 feed through 电压，其次是由前一条 gate driver 走线电压拉回经由 C_s 的 feed through 电压，最后是由本身的 gate driver 走线电压拉回时，经由 C_{gd} 的 feed through 电压的影响。

s 的 feed through 电压，这电压是扮演将显示电极电压推升到正极性电压范围的最重要角色，而最后则是本身 gate driver 走线电压下拉时所产生的 feed through 电压，这个电压由于是经由 C_{gd} 的关系，而且变化的幅度也不大，所以影响也比较小。

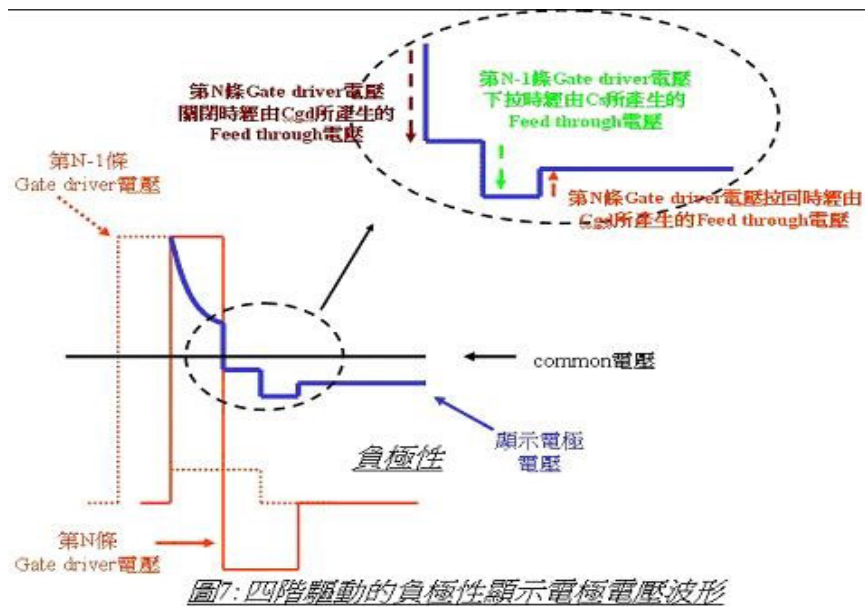


圖7: 四階驅動的負極性顯示電極電壓波形

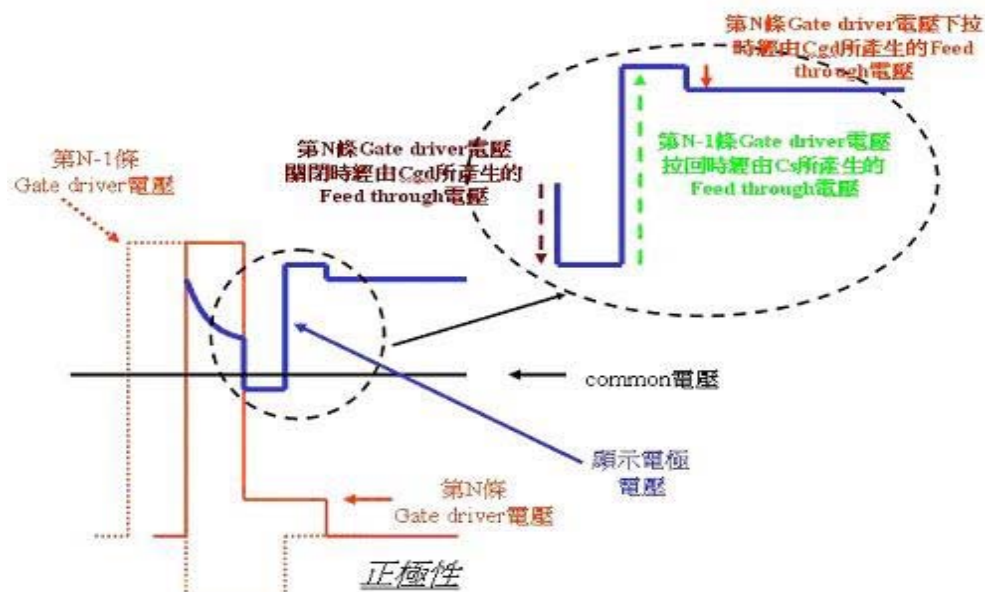
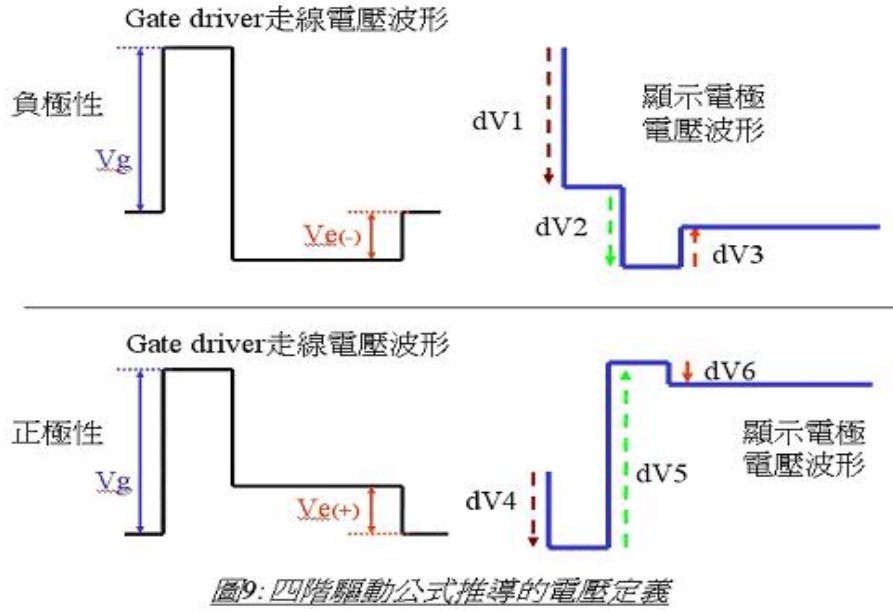


圖8: 四階驅動的正極性顯示電極電壓波形



既然这些电压的操作原理, 都是藉由 feed through 电压的影响, 我们就可以利用计算 feed through 电压的公式, 依照图 9 的电压定义, 来推导出各阶电压的大小. 其结果如下:

属于负极性电压的各个 feed through 电压:

$$dV1 = [Vg + Ve(-)] * Cgd / [Cs + Clc + Cgd]$$

$$dV2 = Ve(+) * Cs / [Cs + Clc + Cgd]$$

$$dV3 = Ve(-) * Cgd / [Cs + Clc + Cgd]$$

属于正极性电压的各个 feed through 电压:

$$dV4 = [Vg - Ve(+)] * Cgd / [Cs + Clc + Cgd]$$

$$dV5 = Ve(-) * Cs / [Cs + Clc + Cgd]$$

$$dV6 = Ve(+) * Cgd / [Cs + Clc + Cgd]$$

在图 6 中, 我们提到补偿后的正负极性输出电压与 common 电压的距离应该一致, 所以给正极性显示电

压范围用的所有 feed through 电压总合应该和给所有负极性显示电压范围用的 feed through 电压总合应该一样。所以 $dV1+dV2-dV3$ 应该等于 $-dV4+dV5-dV6$ 。合并化简后，我们可以得到：

$$V_e(-) - V_e(+) = 2V_g * C_{gd} / C_s$$

也就是说，只要 $V_e(-)$ 与 $V_e(+)$ 的差值，符合上述公式，就可以达到四阶驱动的效果了。而且上述公式也告诉我们一个现象，那就是公式化简后，已经没有了 C_{lc} 的成分存在。因此它跟三阶驱动一样，不会受到 C_{lc} 非线性的影响。至于 $V_e(-)$ 及 $V_e(+)$ 的大小如何决定呢？我们回过来观察 $dV1+dV2-dV3$ 与 $-dV4+dV5-dV6$ 的化简结果：

$$dV1+dV2-dV3 = V_g * C_{gd} + V_e(+)*C_s = \text{向下的 feed through 总量}$$

$$-dV4+dV5-dV6 = -V_g * C_{gd} + V_e(-)*C_s = \text{向上的 feed through 总量}$$

只要我们依照液晶的特性，便可得知需要向上或是向下的 feed through 电压总量需要多少才够（一般是液晶的 threshold 电压与最大工作电压加起来的一半），再参考面板设计的参数 V_g , C_{gd} , C_s 的大小，就可以计算出所需的 $V_e(+)$ 与 $V_e(-)$ 了。在数学上来说，当你把向上与向下的 feed through 总量，都设为 0 时，就可以得到 $V_e(+)= -V_e(-)$ ，这时候四阶驱动就变成三阶驱动了。因此三阶驱动也可以说是四阶驱动的一个特例。

参考数据

·交通大学次微米人才培训课程，平面显示器原理讲义。

·财团法人自强基金会电子工业人才培训课程，液晶显示器显示原理讲义。