NAND Flash 编程问题

DearMs 錢,

由客戶提供的資料看來客戶需要在 Block 0 建立兩組 table 表來管理 bad blocks，因爲 NAND flash memory 的應用方式不同，每個客戶都會有不同的需求；基於某些因素考量，除台灣的客戶我們會視情況以專案方式處理，其他地區我們無法做此支援，很抱歉。
目前 S/W 只提供三種 programming mdoe 給客戶使用，可參考以下資料，目前 M4-NAND-TS48 沒有客戶反應過燒錄無鉛 IC 有問題，所以此 module 並沒有做此區分，有任何其他問題請告知。

To improve yields and keep costs down, NAND devices contain randomly located bad blocks in the array. A programming approach for NAND devices therefore must have a scheme to identify and avoid the bad memory cells when programming the device.

If a memory location is bad, then the entire block is "marked" as a "bad block." Typically, a NAND device starts out with a few bad blocks that are marked by the factory.

To handle these bad blocks in the embedded system, an extra "layer" of software is equired. (This is similar to the software required to manage hard disks that can have bad sectors.)Therefore, to program these devices, all we need to do to handle the bad blocks is to treat them in exactly the same way as they are handled by the user's system.

We have two methods to handle bad blocks. One is to skip over bad blocks and place the data in known good blocks. (We call this Skip Bad Block.) Another is to allocate some of the blocks as a "Reserve Block Area". (We call this RBA Style.)

If your bad block scheme is different form ours, maybe you could use the third way -- Hardcopy to program your device. The device will be the same as the original. Hardcopy is usable only for devices without bad blocks in the range you want to program.

Please confirm the bad block scheme first. If any of the three methods doesn't match yours, you can't use ALL-100 to program your NAND devices. Below are the three methods to handle bad blocks :

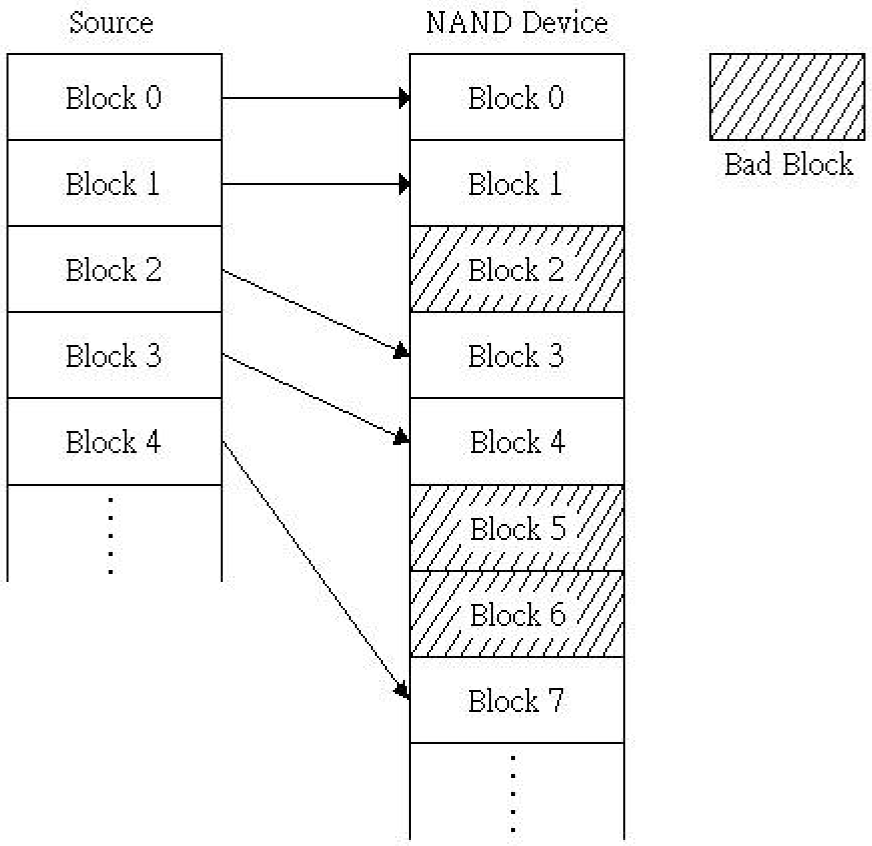

Figure. Skip Bad Blocks

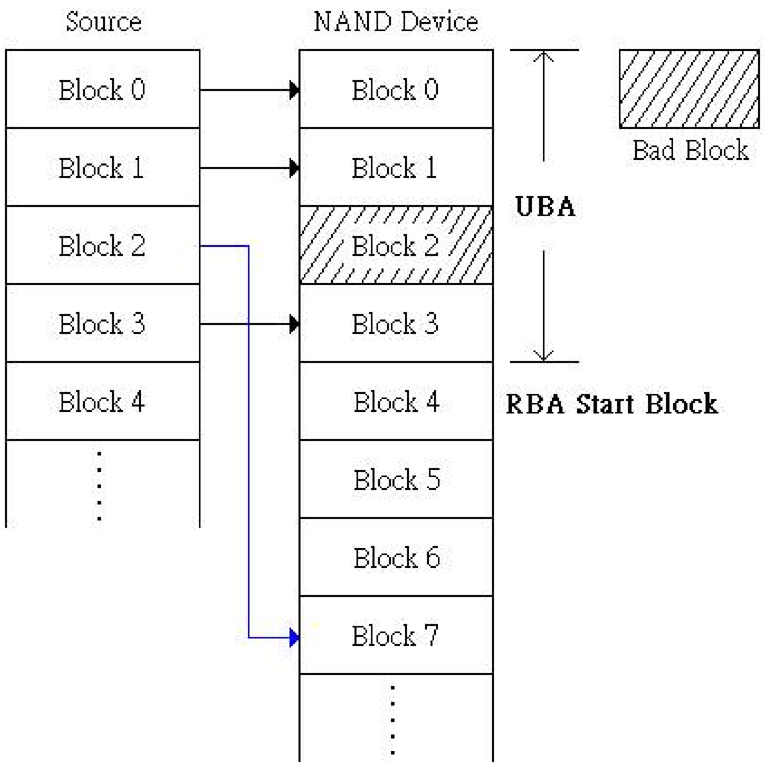

Figure. RBA Style

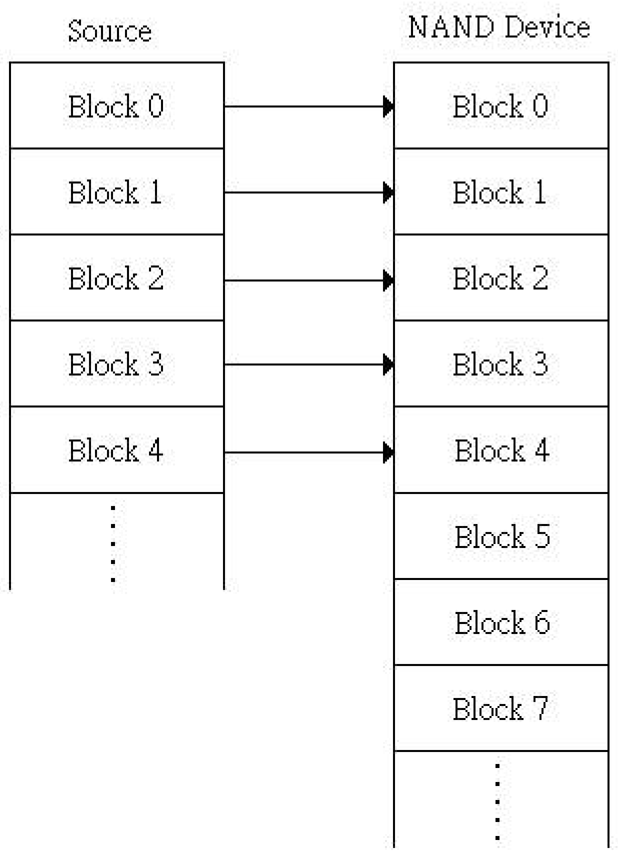

Figure. HardCopy

Best Regards,

Eddie Chung
Technical Support
HI-LO System Research Co., Ltd.