High-Speed Patterning of Hexagonal Dot Arrays Using Electron-Beam Lithography

Bryan Cord, Hans Romijn*, Marc Wijnands*
University of Minnesota Nanofabrication Center, Minneapolis, MN
** Vistec Semiconductor Inc., Watervliet, NY*

A large number of nanofabrication applications in photonics, templated self-assembly, patterned media, and other areas rely on the ability to fabricate arrays of nanoscale dots in a hexagonal configuration over large areas. Electron-beam lithography is one of the key technologies used to accomplish this due to its high resolution, maskless patterning, and wide availability in research environments. Unfortunately, as with most EBL-based processes, throughput remains a serious issue; patterning a 1 cm^2 area with a 100-nm-pitch hexagonal array can take days on even the fastest EBL systems.

We have developed a method, based on exploitation of the pixel-based exposure mechanism of EBL systems, for patterning hexagonal arrays much more quickly than standard writing schemes. The method is based on the popular "pixel-skipping" method of patterning similar arrays in a square configuration. In the "pixel-skipping" scheme, the dot array is defined as one large polygon, which is then fractured into pixels that are much larger than the width of the beam. Exposing this pattern results in a discrete array of dots with a period equal to the pixel size, rather than a continuous shape greatly improves throughput by eliminating intra-dot beam settling time.

This technique, unfortunately, is not directly transferable to hexagonal arrays, since EBL systems write using square pixels arranged in rectangular patterns. In our hexagonal-array scheme, the dot array is defined as two interleaved arrays of one-pixel-wide rectangles as shown in figure 1. Exposure of these rectangles will result in a hexagonal grid of dots, with the dot spacing equal to the pixel size of the rectangles and the dot size defined by the dose used and the spot size of the beam. By carefully controlling the height and width of the rectangle arrays, as well as the mainfield and subfield sizes used to fracture the final pattern, an unlimited number of these arrays can be stitched together to give an arbitrarily large array of dots. As figure 2 shows, the time savings over patterning using explicit dot definition is significant, particularly at denser pitches.

The program we have written takes as inputs the desired size and pitch of the dot array. It outputs a pattern file consisting of a large number of interleaved-rectangle arrays, as well as a set of parameters to be used when fracturing the pattern. The pattern files it generates should work on most fracturing software and EBL systems, although in our case LayoutBEAMER was used to do the fracturing and a Vistec EBPG5000+ was used for the exposures. The micrographs in figure 3 show that the dot arrays patterned with this scheme are very regular, with minimal dot-placement errors despite the lack of intra-dot settling time.

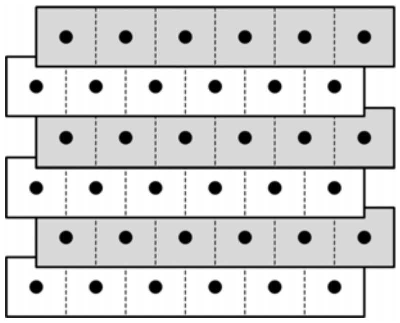

Figure 1: Illustration of the scheme used to quickly pattern hexagonal arrays. The bold rectangles are the shapes used to define the array, with the dotted lines dividing them up into the pixels actually exposed by the e-beam system. The black dots show the position of the dots after the exposure. Dot diameter can be controlled to a fine degree by varying both the beam spot size and the dose per dot.

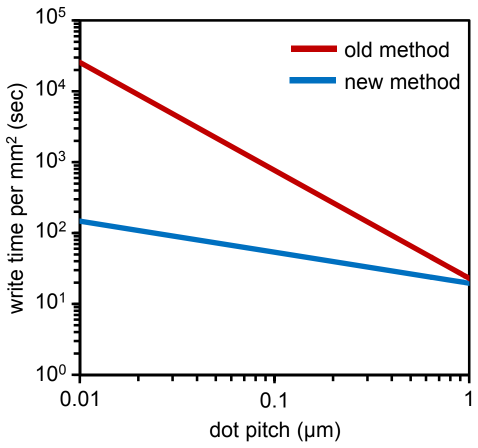

Figure 2: Write times for 1 mm^2 hexagonal dot arrays of various pitches, using both the traditional (each dot is defined explicitly) write method and our improved version, assuming a 50 MHz DAC frequency during the write.

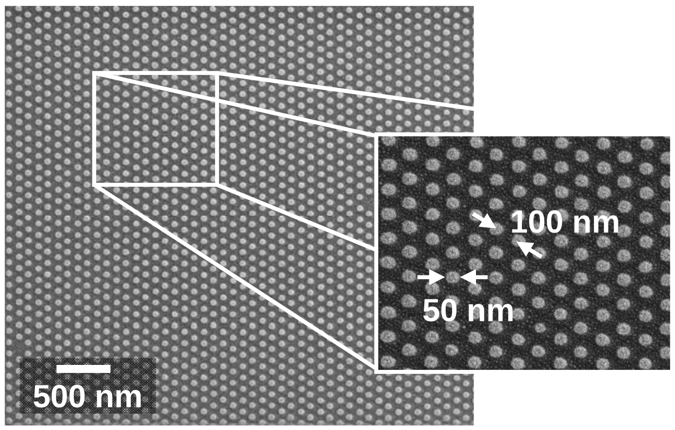

Figure 3: Scanning electron micrograph showing a 100-nm-pitch hexagonal array of 30-nm-thick HSQ dots defined using our scheme. The dots were exposed at 100 keV using a dose of 100 fC/dot and a 10 nA beam current. The write time for a 1 cm^2 array of this density using a typical 1 MHz DAC frequency is approximately 3 hours, while the same array would take ~1.5 days if exposed using the traditional method.