INTEGRATION OF CAD DATA TRANSCRIPTION FOR E-BEAM LITHOGRAPHY.

Ian David Fink
Senior Microelectronic Engineering Student
Rochester Institute of Technology

ABSTRACT

A software system for preparing data for e-beam mask making called Computer Aided Transcription Software is described. In order to appreciate its capabilities, an overview of writing requirements for a MEBES I e-beam tool are given. Sample files and software output are provided to illustrate the concepts.

INTRODUCTION

The method of choice for mask and reticle making for high density VLSI applications is electron beam lithography, due to its high resolution capabilities. E-beam lithography can resolve minimum mask dimensions in the 0.3um range. A second advantage is the ease with which mask and reticle patterns can be modified to support changes in circuit layouts. This is due to the extent of computer control these systems exhibit. These advantages have outweighed drawbacks such as reduced throughput and increased cost, making e-beam the dominant mask making tool.

Writing patterns using electron-beam is controlled though a computer that is tied to the tool. The computer determines the electron beam characteristics (diameter, current, etc...), as well as the movement of the beam and/or stage. In order to write a pattern, the electron beam must be scanned across the substrate exposing all required areas. The two prominent writing strategies for e-beam lithography are raster scanning and vector scanning. Raster scanning involves continuous movement of both the stage and the beam to cover the entire substrate. Typically, the stage is moved back and forth in the x-direction under the e-beam, which is scanning in the y-direction at a high frequency. Between the movement of the stage and deflection of the e-beam, the entire field of the reticle is scanned. Conversely, a system using a writing strategy based on vector scanning would not scan all areas of the substrate. The stage positions the substrate and the beam is deflected, in both the x and y direction, to the areas to be exposed. Once all patterns within the beam's defection range have been exposed, the stage moves to the next zone, and the exposure proces is repeated.

A pattern to be written using raster scanning is divided into stripes, the height of which are determined by the amount of beam deflection. These stripes are exposed and abutted together as the substrate traverses under the beam in a series of passes

as shown in Figure 1(a). The actual pattern to be exposed must be 'fractured' into geometric shapes which can be exposed individually by the scanning beam. These shapes are usually trapezoids which approximate the original layout as closely as possible. Vector scanning also requires a fractured pattern approximated by trapezoids. However, instead of exposing these shapes as they appear in a series of stripes, the beam moves in a vector direction from one trapezoid to the next. These shapes are systematically filled in as shown in Figure 1(b). The information approximating the layout with trapezoids is stored in a 'Pattern File'. This file is used by the e-beam computer to write the pattern.

Figure 1(a): Raster Scanning. (b): Vector Scanning.

The trapezoids in the pattern file conform to a specified fracture grid. This fracture grid serves the same purpose as a resolution grid in CAD circuit layouts; to define the minimum feature size. All trapezoid sizes are multiples of the specified fracture grid structure. It is extremely advantageous to have the fracture grid be the same as the resolution grid that was used to originally layout the design. This facilitates the error free construction of the fractured pattern.

Both raster and vector scanning require the electron beam to be blanked from the photoresist over areas which are not to be exposed. Beam blanking is most often performed by two plates near the electron source at the top of the electro-optical column. A strong positive potential is applied to one plate while a strong negative potential is given to the opposite plate as the beam passes over areas which are not to be exposed. The e-beam is then deflected from reaching the substrate and exposing the resist as it travels between these plates when they are charged.

RIT has recently acquired a ETEC MEBES I e-beam system. MEBES uses the raster scan method, and has the capacity to write reticles and masks, as well as, directly to device wafers. A file, called a job deck, specifies the size of the substrate, as well as the dimension and position of the array to be written. The job deck references pattern files which contain the fractured layout data. In MEBES format six values are neede to specify a trapezoid: x, y, delta x1, delta x2, h, l. These parameters are defined for an arbitrary trapezoid in Figure 2(a). An example of how a MEBES pattern file represents a general shape is given on Figure 2(b).

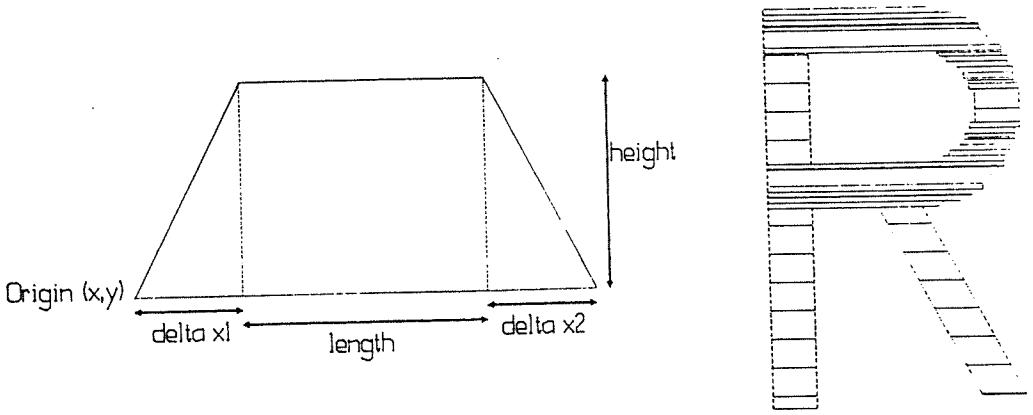


Figure 2(a): MEBES Trapezoid. (b): Fractured Pattern.

The process of fracturing circuit layouts and compiling them into pattern files requires very complex software. The fact that circuit layouts are stored in "standard" files, which may be in any one of several formats (CIF,GDS-II, etc...), makes a versatile software package, capable of deciphering these various files, essential. Computer Aided Transcription Software (CATS) is a powerful software package that can take most common layout formats, fracture the layout, and create output files compatible with an equally impressive number of mask generation tools, including MEBES. This project involved implementing CATS into the mask making process at RIT.

CATS AT RIT

CATS was installed on both the VAX mainframe system and a VAXstation 3100 DEC workstation. The mainframe terminal was utilized to acquire and prepare circuit layout files to be entered into the main CATS program. The actual CATS program used to create fractured pattern files was located on the workstation terminal due to the need for high resolution graphics capability.

Features of CATS include methods of producing fractured pattern files and verifying them against the original layout. Circuit layouts may also be scaled, biased, or reoriented about a specified origin prior to finally fracturing the layout to produce the desired pattern file. Healing functions which reduce the occurrence of errors in fractured designs due to differences between the fracture grid and original design grid, are another important aspect of CATS. These features of the software were tested to verify the CATS' utility to the RIT mask making process.

A second important aspect of CATS software is its ability to construct MEBES job deck files to reference patterns and specify the array of die to be written. This function of the software was also tested. A mask was written with the MEBES using both a job deck and pattern files created using CATS as a true test of the software's performance.

RESULTS/DISCUSSION

Using CATS, all forms of CAD circuit layout files encountered at RIT were successfully fractured into MEBES pattern files with only minor difficulties. The integrated circuit editor (ICE) at RIT is an in house CAD tool used to create circuit layouts. This program saves layouts in CIF format, and was the source of some difficulties. While attempting to enter a standard RIT NMOS cell layout produced on ICE into CATS, it was found that the output files produced by ICE did not follow the CIF convention for defining boxes. ICE CIF files had the order of box position and box size switched. These files had to be edited to define the origin of the box first, followed by the height and width. Once the ICE CIF files were corrected, they were entered into CATS and fractured without any problems. In addition, CATS was successfully used to scale and orient layouts produced on ICE prior to fracturing as desired.

Layouts constructed with Mentor Graphics on the Apollo system at RIT can be saved in either CIF or GDS-II format. A test layout for multi-chip module research, designed at RIT on the Apollo, was sent to the VAX mainframe as a CIF file, and was successfully entered into CATS and fractured to a pattern file. Several other files from outside sources, including a test pattern from KODAK used for focus/exposure characterization of lithography tools, were entered into CATS from magnetic tape. The CAD tools used to create these layouts was unknown, but because they were stored in standard GDS-II format they were fractured without any problems. With CATS, the KODAK layout was scaled five times prior to fracturing for use as a 5X reticle. This also confirmed the modification of layouts using CATS.

The ability of CATS to 'heal' files and eliminate flaws in the fractured pattern, due to differences between the resolution grid of the design layout and the fracture grid used to produce the pattern file, was shown to be very effective. Figure 3 shows a section of a layout exhibiting some of the pattern flaws which may occur. The voids and overlapping are due to the fracture grid not being a multiple of the original design grid. The figures were forced to snap to the fracture grid causing pattern distortion and flaws. The multi-chip module test layout consisted of meandering metal paths. These paths had voids when fractured without CATS healing functions activated, but had no errors when the healing functions were activated.

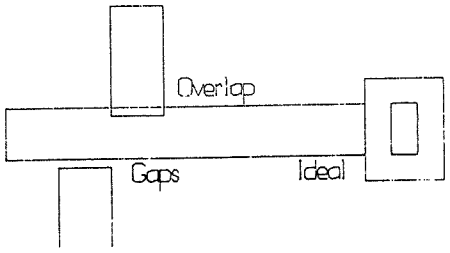


Figure 3: Possible errors in fractured pattern without healing.

Even with pattern healing, fractured pattern files can differ from the layout from which they were produced because the trapezoids in the pattern file only approximate the design layout. For more complex shapes, this approximation is far from perfect, and it is therefore important to know were the fractured pattern deviates from the original layout. This process is called pattern verification.

Fractured pattern files were verified against the original layout using the compare function within CATS. This function compares the fractured pattern file with the input design layout file, and creates a third file displaying all differences. Figure 4(a) shows a focus star from the KODAK layout. Figure 4(b) shows the focus star pattern fractured using CATS. Areas of the layout not perfectly reproduced by the fractured pattern are shown in Figure 4(c).

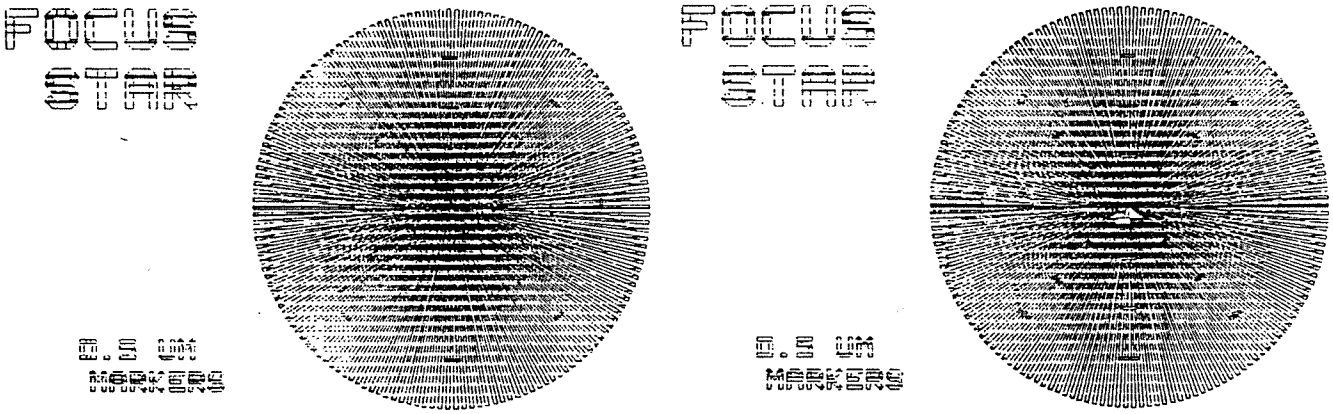
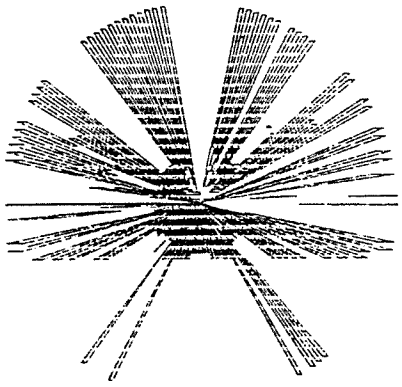


Figure 4(a): Layout. (b): Fractured Pattern.



(c): Differences.

CATS was also used to produce job decks for the MEBES. Decks were constructed for a 5X reticle using the KODAK pattern file, as well as a 4 inch mask using a fractured pattern of a layout from Occidental Chemical. Figure 5 shows the Occidental Chemical Corp mask with the array defined. For each chip in the array the CATS pattern file is referenced and written at that site. The final phase of the project was to actually write the mask using the job deck and pattern file created. This was done successfully, confirming that CATS can be used for all phases of preparing a design layout to be written as a mask pattern.

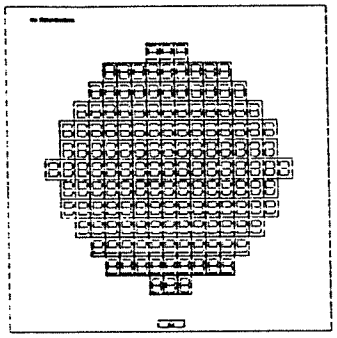


Figure 5: MEBES Job Deck array specification.

CONCLUSIONS

CATS software was used to integrate the processes of modifying and fracturing a design layout to produce a MEBES pattern file, pattern file verification to determine differences between the pattern file and original layout, and creating job decks to define the array to be written with the MEBES. This integration allows CATS to provide a standard method to prepare design layouts for mask making. Because the power and usefulness of CATS was demonstrated this software will continue to play an increasing role in the mask making process at RIT.

ACKNOWLEDGMENTS

Special thanks to Bruce Smith of the RIT Microelectronics faculty, Mark Trembley at the RIT Computer User Center, and Todd Pegelow at Transcription Enterprises LTD.

REFERENCES

[1] CATS Command Reference Manual. Transcription Enterprises LTD. 1990.
[2] MEBES Software Reference Manual. Perkin Elmer.
[3] D. J. Elliot, Integrated Circuit Mask Technology, (McGraw Hill Book Co., New York, 1985), pp 110-122.