A Technology-independent CAD Tool For ESD Protection Device Extraction - ESDExtractor

R.Y.Zhan, H.G.Feng, Q.Wu, G.Chen, X.K.Guan and Albert.Z.Wang
Dept. of Electrical & Computer Engineering, Illinois Institute of Technology
3301 S. Dearborn St., Chicago, IL, awang@ece.iit.edu

Abstract

The challenges for developing an ESD (Electro-static Discharge) layout extractor originate from unconventional layout patterns of ESD protection devices, parasitic ESD device extraction and device count reduction. This paper reports a new technology-independent layout extractor, ESDExtractor, which is capable of extracting all types of ESD devices and answers the demands for ESD design verification. General methodology to extract both intentional and parasitic ESD devices, specific algorithms and implementation methods for efficiency-enhancement are presented, followed by a design example.

1. Introduction

ESD protection circuits are used to prevent ESD damages to ICs [1]. An ESD protection layout extractor is critically important since it makes the further layout-schematic checking and ESD circuit simulation at whole chip level possible.

Compared with vast research efforts on ESD protection circuit design, very limited work has been done for ESD device extraction [2,3]. Current challenges in developing ESD protection device extraction tools follow: Firstly, extraction of ESD device differs from that of normal IC devices in that ESD devices often use unconventional structures. Conventional IC devices, e.g., MOSFET, can be extracted using basic Boolean operation of layout features [4]. However, ESD devices have very complicated and irregular layout patterns, which cannot be extracted using conventional extractors. Secondly, ESD transient current often causes turn-on of parasitic ESD devices existing either inside the core circuit or protection network before an intentional ESD devices can be triggered, resulting in pre-mature ESD failure at chip level. Therefore, parasitic ESD devices must be extracted and analyzed according to ESD stress conditions. The final netlist output should contain all ESD devices possibly being turned-on under ESD stress, intentional or parasitic. [2] presents an ESD CAD tool for detecting parasitic BJTs under ESD stress. In [3], technique to extract arbitrary devices, mainly intentional ESD devices, is discussed. In this paper, ESDExtractor, a new CAD tool, capable of extracting any ESD devices and taking full consideration of intentional and parasitic ESD device extraction and analysis, is presented, with its flow chart shown in Fig.1, consisting of four steps: 1, to read in device definition in technology file and save it in model database; 2, to save layout data into layout database after necessary prepossessing; 3, to recognize all ESD devices, intentional or parasitic, in extraction engine; 4, to check extracted device against parametric and connectivity criteria to remove non-critical ESD devices for count reduction.

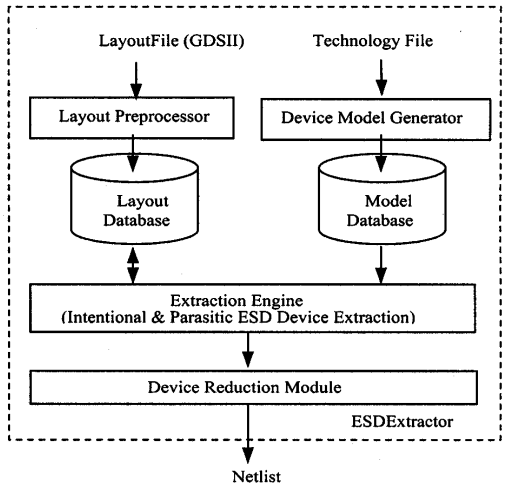


Fig. 1. Flow chart of ESDExtractor

2. ESDExtractor – A New CAD Tool

2.1 ESD Device Extraction Engine

2.1.1 ESD Device Recognition Approach Based on Subgraph Isomorphism Technique

IC layout is a geometrical description of masks in specific formats, e.g., GDSII. Conventional devices, e.g., MOSFET, can be readily presented by a unique set of stacked masks (vertical device). However, ESD devices have complicated irregular layout patterns, typically being lateral devices, which cannot be dealt with by traditional circuit extractors, e.g., popular SCR (silicon controlled rectifier) and LVSCR (low voltage SCR). Apparently, conventional Boolean-based device recognition method [4] cannot recognize ESD devices. We propose to use the subgraph isomorphism technique to realize efficient device recognition of arbitrary ESD devices of any types. For device recognition using subgraph isomorphism, each device type is described by a unique device model graph (*MG*). The whole IC layout data is presented as a target graph (*TG*). A device is then recognized by matching one fragment of TG with the MGs.

Some reported device extractors [3,5] use a top-down approach, where a characteristic feature of a device is specified as the root for *MG*, and the recognition of a device is triggered by discovering its root. This can be an efficient solution when a limited number of device types are considered, but not suitable

0-7803-7607-2/02/$17.00 ©2002 IEEE

as the number of *MGs* increase, as in ESD protection circuit designs. The number of possible mask combinations is limited for a given technology. For example, in n-well technology, active layers are N+, P+, N-well and Poly-Si, so the number of possible layer combinations is 2^4. Considering that N+ and P+ should not appear at the same time in a combination, the number of possible layer combination is 12. If the number of *MGs* is close to or larger than that number, the algorithm becomes inefficient. However, an ESD device extractor has to deal with many device types including classic ESD device types, e.g., SCR, and advanced ESD devices, e.g., a dual-SCR device [6]. Apart from the larger number of *MGs*, another uniqueness of ESD devices is that different ESD devices often share common features. Fig.2 shows that a LVSCR ESD unit shares some common features with a MVSCR (medium-voltage SCR), while a NMOS FET is totally contained by a LVSCR ESD structure. Based upon this observation, we improve the efficiency of ESD device extraction by developing a new device recognition algorithm based on decomposition approach [7]. Unlike the top-down approach reported, which handles each *MG* individually, our algorithm first explores the relationship between *MGs* by a decomposition procedure. The common part of two or more *MGs* will only be matched once, making device recognition not so heavily dependent upon the number of *MG*s as in the top-down approach. In addition, model containment relationship can be explicitly expressed, so the redundant device, can be easily removed from the final result, thus eliminate the necessity to develop an individual redundant device processor.

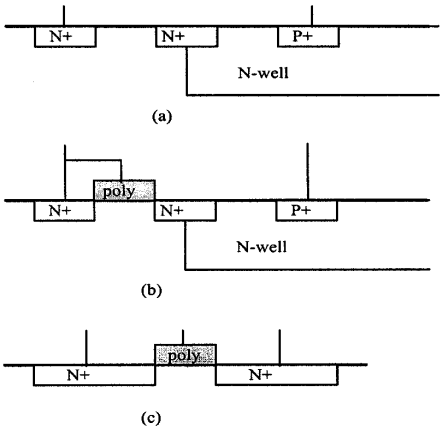


Fig. 2. X-sections of ESD units: a) MVSCR, b) LVSCR, c) NMOS.

2.1.2 Device Model Definition

A non-root graph is adopted for MG definition. Structure of a *MG* is defined as *MG (N, R)*, where *N* is a set of nets in the device and *R* is a set of relationships between nets, including geometrical adjacency *R1* and electrical connectivity *R2*. A *MG* for a simple ESD device, ggNMOS (grounded-gate NMOS), is given in Fig.3, where each circle represents a net, solid lines represent *R1* and dashed lines represent *R2*. The extraction engine aims to extract all possible ESD devices. Intentional devices often obey strict layout rules and utilize some mechanism for triggering or isolation. Fig.4 (a) illustrates a cross section of an intentional SCR featuring isolation guard-ring and pick-ups. However, a parasitic SCR device is irregular in layout with no guard-ring and pick-ups. Accordingly, the dashed box is picked out as generic *MG* for SCR, as shown in Fig.4 (b), so that no parasitic ESD devices will be missed.

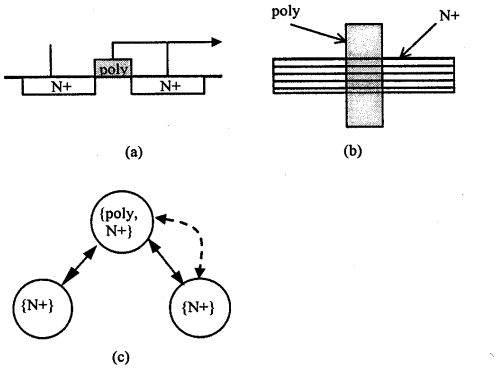


Fig. 3. A ggNMOS ESD device (a) X-section (b) Layout (c) *MG*.

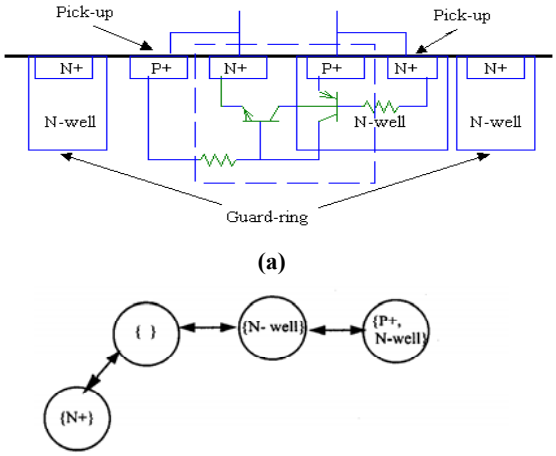


Fig. 4. (a) X-section of an intentional SCR ESD protection structure, (b) the generic SCR *MG*.

2.1.3 Necessary Pre-possessing for Device Recognition

To prepare for device recognition, a preprocessor transforms GDSII layout to *TG* format. A *TG* consists of a set of vertices, each corresponding to a non-overlapping polygon, and a set of edges, representing adjacency between vertices. A scanline algorithm [4] is used in implementation.

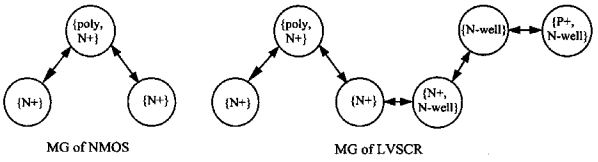
2.1.4 Device Recognition Algorithm Based on Decomposition

This device recognition algorithm is realized in two stages. 1, ESDExtractor decomposes *MGs* recursively and the results, *D(M)*, are represented and saved by an inner data structure. 2, during extraction, the *TG*, is matched with the *D(M)*. As described earlier, when a MG_1 is contained by another MG_2, i.e., a smaller device contained by a larger one, the redundant device corresponding to MG_1 should be eliminated. In our algorithm, the decomposition procedure is also conducted based on the subgraph isomorphism technique. Thus the model containment relationship can be explicitly expressed in *D(M)*, making redundancy checking completed during device recognition and eliminating the need for a redundant device remover.

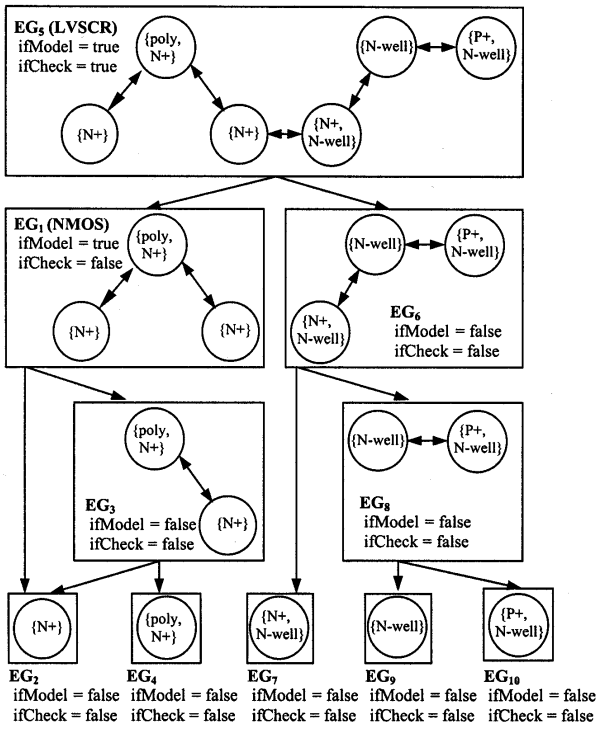
Let $M=\{MG_1, \ldots, MG_N\}$ be a set of device model graphs. The decomposition result of M, $D(M)$, is a finite set of extended graph $EG(G, lG, rG, E, ifModel, ifCheck)$, where

1. G, lG, rG are graphs; lG is left subgraph of G; rG is right subgraph of G;
2. E is a set of edges existing between lG and rG to construct G;
3. $ifModel$ is a Boolean variable. If $ifModel$ = true, G is corresponding to a device model;
4. $ifCheck$ is a Boolean variable. If $ifCheck$ = true, G may contain another MG, directly or indirectly.

Decomposition follows: For each MG, extraction engine first finds the largest subgraph in $D(M)$, $Smax$, then recursively decomposes MG into two parts until reaching individual vertex.



(a)



(b)

Fig. 5. (a) MGs for NMOS and LVSCR, (b) Decomposition results when only NMOS and LVSCR considered.

Fig.5 (a) illustrates the MGs for a LVSCR and a NMOS ESD device. Let MG_1=NMOS and MG_2=LVSCR, the decomposition result, $EG_1 \sim EG_{10}$, in $D(M)$, is shown in Fig.5 (b). As can be seen, NMOS is first decomposed to lG, EG_2 ({N+}), and rG, EG_3 ({Poly, N+} and {N+}). Because lG contains only one vertex, it will not be decomposed. rG will be further decomposed. The decomposition results of the NMOS are saved in $D(M)$ as $EG_1 \sim EG_4$. For LVSCR, the largest subgraph of its MG in $D(M)$, $Smax$, is first detected as $EG_1 \sim EG_4$. LVSCR is then decomposed as lG, EG_1 (NMOS), and rG, labeled as EG_6, the subtraction result of LVSCR by NMOS. Next, it decomposes EG_6, resulting in $EG_7 \sim EG_{10}$. We can see that the common fragment subgraph shared by two MGs, NMOS in this case, is saved only once in $D(M)$. In addition, LVSCR (EG_5), $ifCheck$=true ensures removing redundant device during recognition. Instead of matching each MG individually onto a given TG, the recognition algorithm first finds all occurrences of individual vertices within TG, which are then merged into larger subgraphs until the level of complete MG is reached.

2.2 Device Count Reduction Module

Numerous parasitic ESD devices may be extracted, of which only a handful might be relevant to ESD operation. So, device count reduction is necessary, which will be conducted based upon a new smart parametric checking mechanism, to be discussed elsewhere. Briefly, the followings are considered in the smart parametric checking: 1, ESD-critical key parameters, e.g., gain, triggering voltage, discharging on-resistance, etc, are extracted as terminal parameters. 2, proper criteria are pre-set for them. 3, smart parametric checking is performed for the new ESD netlist generated to remove any non-critical ESD devices extracted. For example, assume two ESD devices, A and B, are extracted from layout and connected in parallel, if device A has a much higher trigger voltage than B does, A will be removed since it will never discharge under ESD. Or else, if device A and B have equivalent trigger voltages, however, A has much less on-resistance than B does, then B will be removed since A will take all ESD transient. This smart parametric checking approach is a major improvement over existing methods where device count reduction are based upon layout dimension only.

3. Implementation And Example

ESDExtractor is implemented in C++ and for Windows and UNIX OSs. It has a user-friendly GUI Interface. The outputs of ESDExtractor are in both text (netlist) and graphical formats. Currently, nine typical ESD protection device MGs, e.g., ggNMOS, ggPMOS (grounded-gate PMOS), SCR, MVSCR, LVSCR, dual-SCR (dual-direction SCR), vertical pnp BJT, lateral pnp and npn BJT are defined in the technology file. Any new ESD structures can be defined when available.

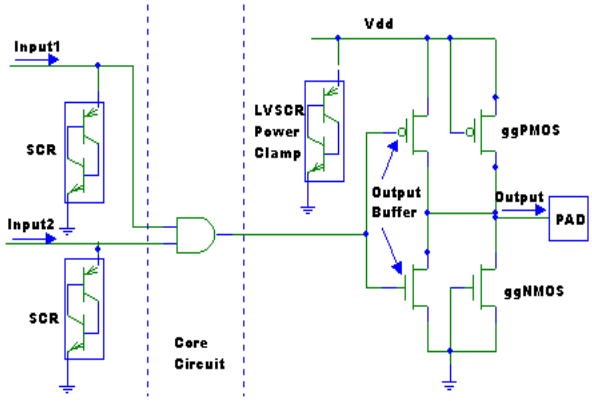


Fig.6 Schematic of an example ESD protection circuit

As an example, a practical ESD protection circuit is used for demonstration. As shown in Fig.6, the schematic, with a symbolic core circuit (an AND gate), uses three types of ESD protection structures, i.e., SCRs at input, ggMOS at output and LVSCR as power clamp. All ESD devices were tested in a commercial CMOS process. Fig. 7 shows the corresponding layout in the GUI of ESDExtractor.

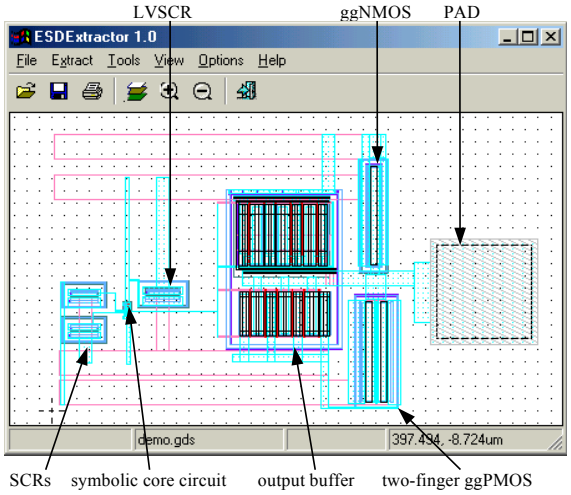


Fig.7 Layout shown in GUI of ESDExtractor

The intermediate netlist report generated before device count reduction is listed below, where each extracted ESD device is presented by its name and terminal nodes in a line. The output file correctly reports all intentional ESD devices, i.e., LVSCR1 for the LVSCR power clamp, SCR1 and SCR2 for the two input SCR ESD protection units, ggNMOS1 and ggPMOS1&2 at output. Note that two ggPMOS devices are extracted for the two-finger ggPMOS structure. Parasitic ESD-type devices, MVSCR1 and SCR3-7, are extracted as well.

```
LVSCR1 Vdd Gnd      (Intermediate text report)
MVSCR1 Vdd Gnd
SCR1 Input1 Gnd
SCR2 Input2 Gnd
SCR3 Vdd Gnd
SCR4 Vdd Gnd
…
SCR7 Vdd Gnd
ggNMOS1 Output Gnd Gnd
ggPMOS1 Output Vdd Vdd
ggPMOS2 Output Vdd Vdd
```

Next, the device reduction module of ESDExtractor executes to eliminate non-critical ESD-type devices using pre-set parametric and electrical connectivity criteria. In this example, ESD-critical parameters extracted indicate that parasitic MVSCR1 of the LVSCR power clamp and parasitic SCR4-7 of the core circuit have much higher trigger voltage than the pre-set criterion value and can be removed. However, parasitic SCR3 of the output buffer unit has a triggering voltage close to LVSCR1 nearby. Further, since parametric and connectivity checking indicate that SCR3 is in parallel with and has a discharging resistance comparable to the intentional LVSCR power clamp, potential risk exists for parasitic SCR3 to compete with the LVSCR, resulting in possible ESD accidental triggering and early damage. Hence, parasitic SCR3 should be retained in the final netlist, as listed below, with all intentional ESD devices, for further ESD design verification.

```
LVSCR1 Vdd Gnd              (Final text report)
SCR1 Input1 Gnd
SCR2 Input2 Gnd
SCR3 Vdd Gnd
ggNMOS1 Output Gnd Gnd
ggPMOS1 Output Vdd Vdd
ggPMOS2 Output Vdd Vdd
```

A graphical output is also generated in GUI with all the intentional and parasitic ESD devices identified by frame-markers, as shown in Fig.8. The run time of the whole extraction procedure of this example from prepossessing to GUI refresh is about 8.2s on Windows platform with 128MB memory.

4. Conclusion

In conclusion, we report a new technology-independent CAD tool, ESDExtractor, for accurately extracting any types of ESD protection devices. The decomposition-based subgraph isomorphism techniques used make ESDExtractor a powerful and efficient extractor. A complete whole-chip ESD design verification CAD package, ESDcat, including ESD device model extraction and ESD simulation based chip verification at both schematic and layout level is under development.

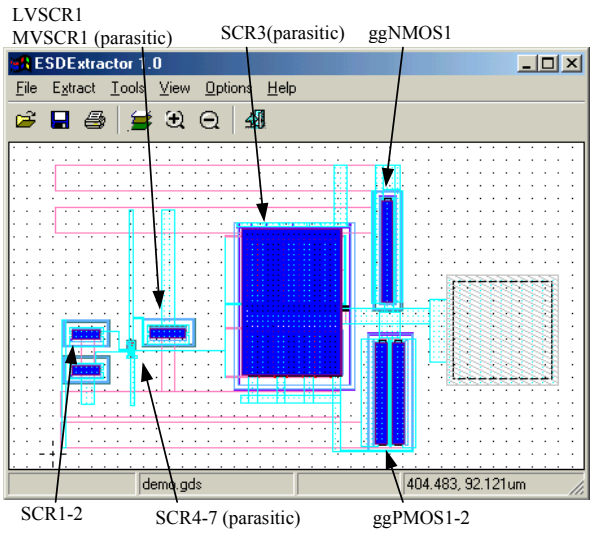


Fig.8 Graphical output illustrates all device extracted.

References

[1] A. Wang, *On-Chip ESD Protection For Integrated Circuits*, Klumer Academic, 2002, ISBN: 0-7923-7647-1.

[2] T. Li, *et al*, *ISCAS'98, vol.6,* 1998, pp389.

[3] Q. Li, *et al*, *Proc. 10th Great Lakes Symp. on VLSI*, 2000, p143.

[4] U. Lauther, *Proc. 18th DAC*, 1981, p555.

[5] A. Brown, *et al, Com., Speech & Vis, IEE Proc., v135 n6,* 1988, p141.

[6] A. Wang, *et al*, *IEEE T-Elec. Dev., v48, n5,* 2001, p978.

[7] B. Messmer, *et al*, *IEEE T-Know. Data Eng, v12 n2,* 2000, p307.