LIBRARY ADVISOR - AN INFORMATION MANAGEMENT SYSTEM ON STANDARD CELL LIBRARY

Lim Wei Pin[1,2], Kuay Chong Lee[2], Lim Kiang Leng[2] and Nurul Hashimah Ahamed Hassain Malim[1]

[1]*Universiti Sains Malaysia,{limweipin@student.usm.my,nurulhashimah@ usm.my}*
[2]*Intel Microelectronics (M) Sdn Bhd, Malaysia, {chong.lee.kuay@intel.com,kiang.leng.lim@intel.com}*

ABSTRACT. Integrated Circuit (IC) Design has always been a challenge to designers. There are two approaches that can be used to draft architecture of an integrated circuit which are the old approach of using Custom and a new one using ASIC. ASIC has been preferred by IC designers as it manipulates liberty files to provide suggestion on the architecture design faster than the Custom approach that relies on tailor-made designs. Tailor-made designs are difficult to draft and not flexible nor scalable. In fact, the time required for an IC to be build based on Custom design is longer than the one using ASIC. Furthermore, ASIC works on a higher-level of the IC Design in which liberty libraries can be sought to find suitable cells for the IC. However, these liberty libraries are huge that it is quite impossible to browse through manually for intensive comparisons to be made. This has been the obstacle in the ASIC design. Therefore, we proposed a system that stores these liberty libraries as a database and is able to intelligently suggest possible candidate components for IC design sought by the system user. This is a new and novel system that would benefit all designers in the IC design field.

Keywords: integrated circuit design, standard cell library, information management system

INTRODUCTION

There are two fundamental approaches to design Integrated Circuit (IC). IC Designers are slowly migrating from Custom to ASIC design. ASIC design is of a higher level and easy to manage. It is a new approach and can be improved through the system. Custom design is of a lower level and more efficient but difficult to produce. Although there are semi-custom or custom in ASIC design (Dally and Chang, 2000), changing design architecture is often more difficult. Thus library file would be the target of the system. Custom design which is more difficult to produce is now slowly replaced by ASIC design. Standard cell library which refers to a collection of liberty files or .lib file is the core of ASIC design. With standard cell library, development time for IC design is greatly reduced from around 2 years to 6 months. This is because designer can purely rely on comparing and analyzing liberty files rather than physically testing the design.

Currently there are not much interactive ASIC standard cell library cross referencing and summarizing information system in the industry. Problem occurs when designers have to choose the correct cell into chip design and designers facing difficulty in analyzing thousands of cell across different process, voltage and temperature, thus trial and error is the only ways

to choose the best cell. Dozens of engineer's resources are needed as the design is going to be larger and harder (Barrett,1976). Beside there are also different fabricators who are the maker of the chip that will provide designers the .lib file. Hence, designers will need to sought through various sources of libraries to know which library is better and more efficient. Currently there are not much ways or tools to compare the library file, and often those existed are very expensive and whose format does not suit to the information needed by designer resulting to the reliance on the less accurate information.

Therefore, we proposed a system called Library Advisor that provides designers the capability to quickly navigate and analyse the strengths and weaknesses across all standard cells, and enable them to select the best fit standard cell to fulfill each individual design challenge. The system greatly reduces the design time and increases the efficiency of design work, hence making sure the design quality is maintained. The proposed system is Unix-based and the standard cell library used is NanGate 45nm open source liberty (Knudsen, 2008).

The library files are stored in a database. This enables user to easily enter, modify or retrieve the information. Analysis can then be done in the database, by slicing correct data and providing good data visualization method. The data analysis method is a hierarchical method where the bottom layer is the data, middle layer is the analysis structure and the top layer is the semantic layer or visualization layer (Lv and Zhao, 2014). A project or system must fit corporate strategy and business objective (Weir et al., 2003). This is extremely important in designing a system that provides viable solution to solve real world problem.

EXISTING APPLICATIONS AND UNIQUENESS OF PROPOSED SYSTEM

One of the application exist in the market is NanGate Liberty Analyzer provided by NanGate that works to optimize standard cell libraries. The Liberty Analyzer is one of the complete solutions available that is able to compare liberty (.lib) files and report the analysis on timing, power, noise and area between libraries (NanGate.com, 2014). Note that these libraries are the open cells liberty files produced by fabricator as mentioned earlier. Each cell will have its own liberty file. Another existing system is called Cadence Virtuoso Liberate that provide an ultra-fast library characterization solution for standard cells, I/Os, and complex multi-bit cells supporting advanced timing, power and signal (Cadence.com, 2014). Cadence Liberate is used to compile the power, timing features of the cell in a timing library. It is useful for the designer to directly read the timing library instead of re-extracting the individual standard cells.

The existing tools mentioned served different usage. Liberty Analyzer is more on general library comparison; it facilitates the comparison of library and provides report in different format such as HTML, CSV and Text data. Whilts, Cadence Liberate, is used for cell characterization rather than overall library comparison. Library advisor provide the best of both world by having comparison and analysis on liberty files and also cell information.

Liberty Analyzer (left column in Figure 1) is dynamic and it only allow two library files comparison. Library Advisor is static and allow more than 2 library comparisons. Let say there are 3 library files A,B and C. Liberty Analyzer can only made pairwise comparison hence resulting a total of 3 comparison of Library A to Library B or Library A to Library C or Library B to Library C. It also need to parse 2 times Library A in order to have different comparison of Library A. With Library Advisor (right column in Figure 1), it provides a static database model where we only need to parse the library once to the database, then any type of comparison can be made based on the database.
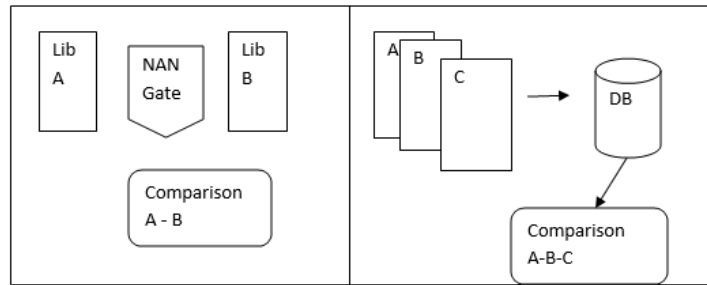
Figure 1. Conceptual comparison of Liberty Analyzer and Library Advisor.

Figure 2. Example of a liberty file.

ARCHITECTURE OF LIBRARY ADVISOR

Library Advisor could be designed in both GUI and non-GUI (terminal) mode. The system was able to accept an argument in the terminal and return a list of cells that fulfill the user's input. It is good to have terminal mode as engineers that could get their desired output with just a line of command and they could adapt Library Advisor to their flow in the future. Library Advisor (as shown in Figure 3) is divided into four core modules that are Library Extraction Modules, Cell Catalogue Modules, Data Analysis & Comparison Modules and Reporting Modules.

Figure 3. Architecture of Library Advisor and its modules.

Library Extraction Module

Library Extraction module was responsible for the large dataset of liberty files. It was further divided into two sub-modules namely the Library Parser module and Library SQL model module. Library Parser module handled the extraction of valuable information inside liberty file. Information such as timing, performance and area are extracted by the system. There were plenty of library file and the total size of the files were more than 100gb. The

speed and efficiency would be an issue to the program. In this module, each liberty file extraction was independent from each other and thus parallel programming could be used to increase the speed of the extraction. Next the Library SQL model module was to parse the extracted information into MySQL database. Further indexing was needed for faster query and access to the database. Indexes were used in MySQL to find rows with specific column values quickly. Without an index, MySQL must begin with the first row and then read through the entire table to find the relevant rows. The indexes used was the B-tree data structure.

Cell Catalogue Module

In this module, user was able to query a list of cells based on certain parameters. Designers could key in several input which was needed to be archived in their design. The program would be able to use KD-tree algorithm to find out the nearest matching cell in the database. Designer can now choose the best fitted cells into their design.

Data Analysis and Comparison Module

Data extracted was analyzed and compared. Engineers could easily choose the version of the library or the type of cell that they are interested in to be compared. Single command was needed to be keyed in by the designer to get the result. Designers could easily switch their preference of comparison without any hassle.

Reporting Module

The reporting modules were used to create tabular and visualization reports. A graph could be plotted from a list of cells or library to make comparison. Report generated was based on user preferences. The generated report was in html format so that it could be easily read by the user. The report would also act as a quality assurance tool for the library team to ensure the library to be released is of good quality.
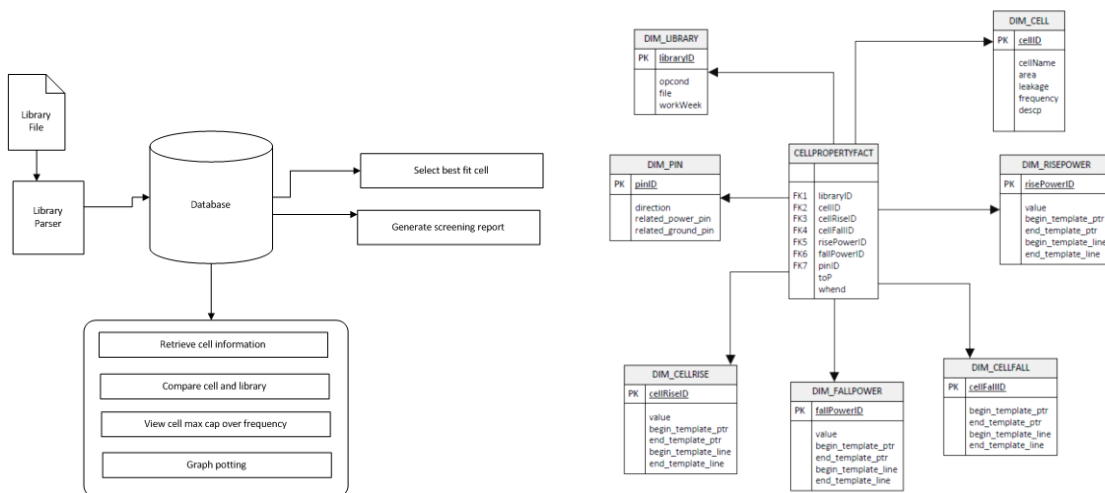
DESIGN IMPLEMENTATION

Figure 4. Library Advisor Architecture and Database Schema.

Figure 4 shows the architecture and database schema of Library Advisor. There are four main components and one database in the system. The four main components' module is Library Extraction Module, Cell Catalogue Module, Data Analysis and Comparison Module and

Reporting Module. Library Extraction Module extracted the information from the library file and loaded into databases. The other three modules then used the information in the database as an input and provided analysis and results to the users.

As discussed earlier, Library Extraction Module was used to extract the valuable information in the library file into the database. The reason for doing this was because there were a few hundred gigabytes of library files that we needed to analyze. By loading it into the database, it could then be easily used by the other three modules, else whenever we needed to analyze or retrieve the cell information, we would have had to look through the library again and this would have consumed a lot of time. Next in Cell Catalogue Module, there were a thousand plus cells in a library file. Since all the cell information resided in the databases, thus user could key in parameters such as 'cellrise', 'cellfall', 'risepower' and 'fallpower' to determine the best fit cell inside the database. The algorithm used here was KD-tree algorithm. KD-tree is a space partitioning data structure for organizing points in a k-dimensional space. This was the earliest data structure proposed for a search involving multidimensional search key (Friedman et al., 1977). It allows us to quickly find approximate nearest neighbors points.

In Data Analysis and Comparison Module, the database could then be used again. User could easily retrieve the information in the library file by giving the input of the cell and library name that they wished to look at. Instead of just displaying the normal information to the users, the system would able to precompute the average value and tell the user what is the best and worst performance and power value. User couldn't easily understand and compare the value with the other version of the cell or library. The user could also easily revert to the original file to know where was the value coming from. Besides allowing the user to plot several graphs based on the information and discover the relationship between each version of cell or library, the user was able to select the max cap over frequency graph which was an important graph for them to understand which cell was unable to operate in high frequency in the library. This is important as if the cell was unable to operate in high frequency but is being used in the design, then it would cause a lot of trouble to the designer in later stage.

The last one which was the Reporting Module, was used as a screening tool for the library. An HTML report was generated based on several values in the database. This report showed the difference between the values in the golden and newest version of the library. It provided a screening capability for the library to ensure the quality of the library before it was released.

Database implementation

Database in Library Advisor played a very important role in the system. The database design here was not a normal relational database design but was a data warehouse design. The design methodology used here was Kimball's Business Dimensional Lifecycle. Kimball's methodology uses new methods and technique in the development of data warehouse (Kimball et al., 1998). It starts by identifying information and requirements and first create a data mart. Data mart is critical in setting the scene for the later integration of other data marts as they come online. It uses dimensionality modeling to establish the data model for each data mart (Kimball and Ross, 2002). The advantages of Kimball's Business Dimensional Lifecycle are scaled down projects which means the ability to demonstrate value is more achievable within an allocated time period or budget. The reason for using data warehouse design rather than relational database design was because data warehouse database is good for the static data and application which does not require much modifying or inserting of data. In Library Advisor, data was loaded into the database when there was a new version of the library file, and after it had been loaded into the database, Library Advisor would not modify the data, instead it would only retrieve the data and perform analysis on the data. The Extraction,

Transformation and Loading (ETL) process was only done possibly once a month.The ETL process was executed in Library Extraction Module. A library parser was responsible to parse the library file, extract valuable information and perform a series of transformation to fit into the databases design. After transformation, the data was loaded into the database.

KD-Tree algorithm

In Cell Catalogue Module, the user keyed in several inputs and got the best fit cell that matched the nearest to its input. There were different nearest neighbor algorithms mentioned in various articles (Bentley, 1975; Arya et al., 1994; Kibriya and Frank, 2007) but mostly merith highly and used the KD-tree algorithm. A k-d tree, or k-dimensional tree is a data structure that is used to organize several number of points in a space with k dimension. It is a binary tree with some constraints imposed on it. The KD-tree used in Library Advisor was in four dimensions; the cellrise, cellfall, risepower and fallpower values. Each level of a k-d tree (see Figure 5) split all children along a specific dimension, using a hyperplane that was perpendicular to the corresponding axis. At the root of the tree all children were split based on the first dimension. Library Advisor is in a 4-dimensional tree, the root would have a w-aligned plane, the root's children would both have x-aligned planes, the root's grandchildren would all have y-aligned planes, the root's great-grandchildren would all have z-aligned planes, the root's great-great-grandchildren would all have w-aligned planes, and so on. It cycles through the axes to select the splitting plane.
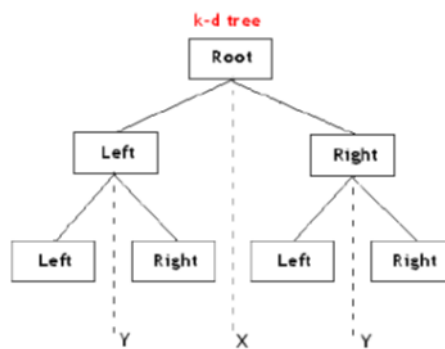

Figure 5. Example of Library Advisor in GUI and Terminal Mode.

RESULT

Figure 6 shows the interface of Library Advisor in both GUI and non-GUI mode. In GUI-mode, it provides several tabs and easy access to basic information that user needed. In terminal-mode, user will have more function and flexibility to choose what they wish to accessed to. Terminal mode would favor more to advanced user.
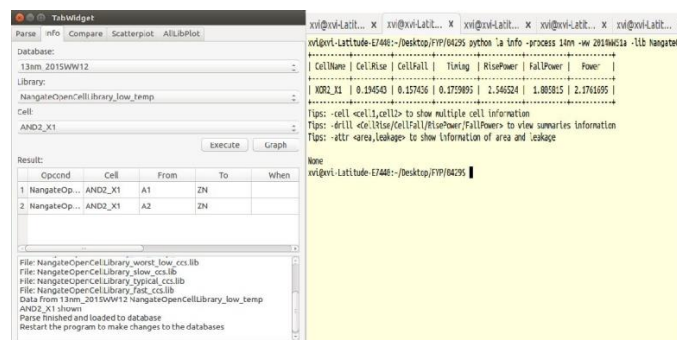

Figure 6. Example of Library Advisor in GUI and Terminal Mode.

151

CONCLUSION

In conclusion, Library Advisor is able to bring benefit to the user. It does not only provide analysis of library but also the screening capabilities of the system would be able to maintain the quality of library and prevent any serious issues from happening inside the library file. By extracting all the information of liberty to database, navigation through the data becomes easy and fast. By having API connected to the database, the liberty data can be easily accessed by other programs, the user can call the API and directly access the database. Some future work suggested is to provide a better and interactive diagrams and reports for ease of analysis.

REFERENCES

Arya, S, Mount, D.M, Netanyahu, N.S, Silverman, R, and Wu, A. (1994). An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms* (SODA '94). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 573-582.

Barrett, W. (1976). An information management system for integrated circuit devices. *IEEE Transtions On Engineering Management*, *EM-23*(4), 147-152.

Bentley, J.L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9, 509-517.

Cadence.com, (2014). *Cadence Virtuoso Liberate Characterization Solution*. [online] Available at: http://www.cadence.com/products/cic/liberate/pages/default.aspx [Accessed 9 Nov. 2014].

Dally, W.J and Chang, A. (2000). The role of custom design in ASIC Chips. In *Proceedings of the 37th Annual Design Automation Conference* (DAC '00). ACM, New York, NY, USA, 643-647.

Friedman, J.H., Bentley, J.L., and Finkel, R.A,. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw. 3*, 3 (September 1977), 209-226.

Kibriya A.M., Frank E. (2007) An Empirical Comparison of Exact Nearest Neighbour Algorithms. In: Kok J.N., Koronacki J., Lopez de Mantaras R., Matwin S., Mladenič D., *Skowron A. (eds) Knowledge Discovery in Databases*: PKDD 2007. PKDD 2007. Lecture Notes in Computer Science, vol 4702. Springer, Berlin, Heidelberg

Kimball, R. and Ross, M. (2002) *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edition*, Wiley, New York.

Kimball, R., Reeves, L., Ross, M., and Thronthwaite, W. (1998) *The Data Warehouse Lifecycle Toolkit*. Wiley, New York.

Knudsen, J. (2008). *Nangate 45nm Open Cell Library*, CDNLive, EMEA.

Lv, M and Zhao, H. (2014). Research on multi-dimensional data visualization analysis of large enterprise's purchasing fund, *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI),* Qingdao, pp. 213-216.

NanGate.com, (2014). *Liberty Analyzer - Standard Cell Library Productivity Tools - NanGate*. [online] Available at: http://www.nangate.com/?page_id=272 [Accessed 9 Nov. 2014].

Weir, R., Peng, T. & Kerridge, J. (2002). Best practice for implementing a data warehouse: a review for strategic alignment. *In Data Warehousing and Data Mining Workshop* (August 2003).