Merging PDKs to build a design environment for 3D circuits: methodology, challenges and limitations

O. Billoint, K. Azizi-Mourier, G. Cibrario, D. Lattard, M. Mouhdach, S. Thuries, P. Vivet
Univ. Grenoble Alpes
CEA, LETI, DCOS, LMDK
F-38000 Grenoble
olivier.billoint@cea.fr

***Abstract*—Design of 3D ICs is mainly done in separated design environments for each tier, assuming that communication channels between tiers are user-defined and fixed at the beginning of the design process. Suitable for 3D stacking based on TSV or Hybrid Bonding technologies because of low granularity of these 3D interconnect elements, this methodology becomes less effective for 3D sequential technologies once trying to integrate Monolithic Inter-tier Vias (MIVs) with higher density (around 1.10^8 vias per mm^2). In this paper, we describe a methodology to create a unified design environment for 3D sequential technology by merging Process Design Kits (PDKs) of different technologies attached to different tiers. Main advantage of this methodology is that designing a 3D circuit may no more require several design environments, thus simplifying simulations, verifications and layout finishing. As a proof of concept, we designed and taped-out a RISC-V processor with logic on memory architecture using LETI CoolCube 28nm FDSOI on top of ST 28nm FDSOI technology.**

Keywords—PDK, Addon, CoolCube, Sequential 3D

I. Introduction

3D stacking technologies such as TSV or Hybrid Bonding have led to major improvements in IC performances and integration density, especially on the memory [1] and image sensor [2] markets. As size of these 3D interconnects can reach around 1µm, they can be considered big enough to be pre-placed and fixed during floorplanning [3], thus simplifying place and route steps which can be separated for each tier (assuming that communication channels created by these 3D interconnects are correctly modeled and integrated in the flow). Considering 3D sequential technology which can reach a 3D interconnect density of 1.10^8 vias per mm^2, the aforementioned design strategy may no more be optimal as it will not take advantage of the placement flexibility brought by the small 3D interconnect size. As sequential 3D digital place and route tools and methodologies are rising [4], the need for a unified multi-tier design environment covering both analog and digital with good quality of results makes more sense. This implies merging files from each PDK (each tier being possibly designed with a different process technology thus using a different PDK) in order to avoid designing in two different environments in the case of a two tier 3D circuit. Our main objective in this study was to implement a consistent design and verification platform including schematic, layout, digital place and route (including lef, lib and techfiles for BEOL parasitic extraction), simulation and verification (full process DRC for each tier, LVS). Our work is based on industry's standard softwares which consists in a cadence Virtuoso design framework, using Mentor Eldo simulator, Mentor Calibre for DRC&LVS, Synopsys Star-RCXT for parasitic extraction, Synopsys Design Compiler for digital synthesis and Cadence Innovus for digital Place&Route. The paper is organized as follows: section II introduces briefly 3D sequential process, PDK content and challenges/limitations of PDK merging, section III describes the modifications done on the analog part of the PDK, section IV details the modifications done on the digital part of the PDK and finally section V gives some information about our use case: the implementation of a RISC-V processor with SRAMs on the bottom tier and digital logic on the top tier.

This work was partly supported by the European Union Framework Program H2020 under project name 3D-MUSE (780548).

II. 3D Sequential Process and PDK Content

3D sequential technology [5] may offer new opportunities to the design community but still suffers from lack of EDA tools especially on the digital design side. From the process point of view, it consists in building a low temperature process including Front-End Of Line (FEOL) and Back-End Of Line (BEOL) on top of a foundry process. Main advantage of this integration is that distance between bottom tier last metal layer and top tier first metal layer is very small (hundreds of nanometers) compared to other 3D integration techniques, which allows small and fine pitch MIVs. This particularity, increasing MIV placement flexibility, can be very useful if accessible in a unified design environment where MIVs could be part of a single BEOL stack going from bottom tier FEOL to the latest top tier metal layer. Figure 1 shows an example of a 3D sequential process stack.

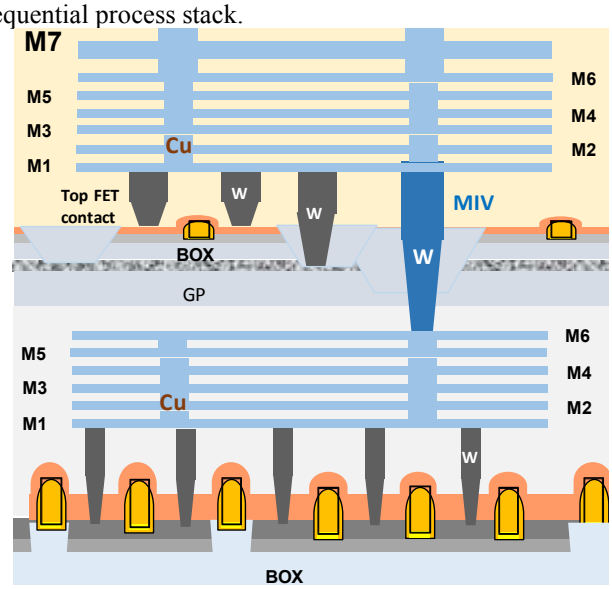

Fig. 1. Example of a sequential 3D SOI process stack (6 metal layers on the bottom tier and 7 metal layers on the top tier). Cu means Copper, W means Tungsten and GP means Ground Plane.

978-1-7281-4870-0/19/$31.00@2019 IEEE

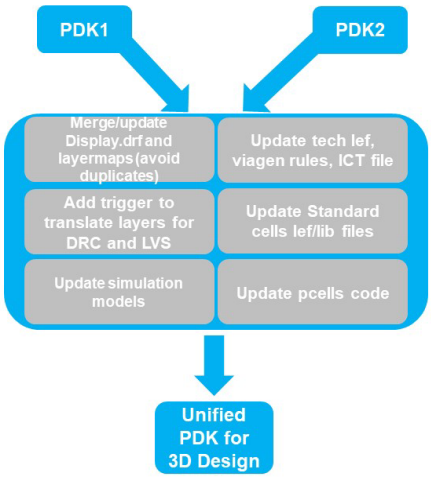

Fig. 2. Methodology to merge two PDKs into a single 3D PDK.

A. PDK Architecture

Once installed, PDKs are usually composed of several folders in which user can find what they need depending on the EDA tools they are using. This goes from transistor models for analog simulation to configuration files for digital place&route, verification files such as Design Rules Check (DRC) and Layout Versus Schematic (LVS) decks or Design Rules Manual (DRM). Depending on the foundry, almost all tools are integrated into a design framework to take advantage of a user-friendly graphic interface. Even if folder names and organization is different from one foundry to another, content is pretty much the same. Added to this PDK, user has access to some digital standard cells libraries containing at least layout views, transistor netlists or schematic views as well as .db, .lib and .lef files for digital place&route.

B. Challenges

Merging two PDKs of possibly two different technologies requires several steps as shown on figure 2 in order to ensure: (1) usability for the designer (especially regarding configuration of the display in the layout window to show all accessible layers of both technologies while discriminating groups of top and bottom layers. (2) Consistency of verification steps (DRC, LVS) and GDS numbers for all process layers of both technologies while avoiding any duplicates (which could be deadly, top and bottom tier technologies being possibly processed in a different foundry). (3) Manufacturability of the 3D interconnect with respect to both top and bottom process technologies. These three main topics: usability, consistency and manufacturability are addressed throughout following sections III and IV for both analog and digital parts of the design environment. It is important to notice that the unified PDK requires correct installation of both PDK1 and PDK2.

C. Limitations

Limitations of PDK merging have two major reasons: the first one being EDA tool capabilities, second one being confidentiality as some configuration files are only available in an encrypted version. This implies that right now, some usual design tasks can't be done in a unified environment but only in a two-step mode. Analog parasitic extraction is one of them because configuration files (like .nxtgrd for Synopsys Star-RCXT) are encrypted as they contain a lot of critical process information. Moreover, extraction of a design including two FEOL and two BEOL while taking into account all coupling effects between top and bottom tier seems not yet resolved. It is important to notice that once a foundry will cover the entire process flow of fabricating and assembling both tiers, all the aforementioned limitations should disappear. Full 3D analog parasitic extraction and 3D full place&route will however require major tool updates in order to take full advantage of the 3D process.

III. Analog Section of the PDK

The analog section of a PDK mainly contains simulation models, pcells, graphic customization of the design framework, verification decks such as DRC and LVS and parasitic extraction deck. State of the art [6] already describes customization required to plug an inhouse addon on a PDK, we'll explain here how we merge two PDKs, possibly from two different foundries.

A. Customization of the Design Framework

In order to customize the Cadence Virtuoso Design Framework with two PDK information, we start by dumping both top and bottom tier Cadence Tech files by doing CIW > Tools > Technology File Manager > Dump in Virtuoso. Once this done, we add a few lines of code to link these two tech files while ensuring no duplicates are created, especially regarding layers and associated purposes. More precisely, according to Incremental Technology DataBase (ITDB) standard bottom tier tech file remains unchanged while top tier tech file is updated and included in the first one as an addon.

Next step is to modify the Cadence Virtuoso layer map and display.drf file to allow a correct and user-friendly display of all layers while ensuring correct import/export of designs for both technologies. For the virtuoso layer map, this is done by script in order to avoid name duplicates as well as gds layer number duplicates (usually adding a fixed offset value to layer numbers). For the display.drf, a script is used as well to create new packets associated to modify layer names and offering the possibility to associate new stipples to layer names. From graphical point of view, stipples customization is very useful to allow an easy discrimination of identical layers on different tiers such as Poly, Active, etc. In our case we customized stipples in order to display a "cc" shape on all top tier layers (cc standing for CoolCube), stipple code is written below and graphical result is shown on figure 3.

```
( display     stipple35_CC      ( ( 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
                                  ( 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 )
                                  ( 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
                                  ( 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
                                  ( 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
                                  ( 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 )
                                  ( 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
                                  ( 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 )
                                  ( 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 )
                                  ( 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 )
                                  ( 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 )
                                  ( 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 )
                                  ( 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ) ) )
```

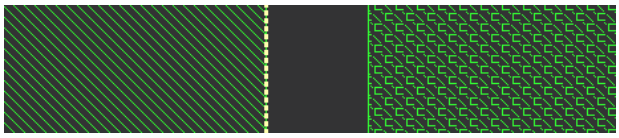

Fig. 3. Example of graphical customization using stipples.

To keep a user-friendly layout window, we added Calibre menus in the layout window to allow launching Calibre DRC and LVS for both tiers in one single run with a unified display of check results. This is done by customizing a file, which is defined by the environment variable MGC_CALIBRE_VIEWER_MENU_CMDS as described in Calibre documentation.

In our case, the inter-tier via is defined like a regular BEOL via which means it has no specific instance from the schematic point of view (like any other via in the technology). From the electrical simulation point of view inter-tier via can be modeled by a resistance of appropriate value, which can fix the lack of full stack Analog Parasitic Extraction during back-annotated simulations, as we will discuss in section E.

B. *Simulation Models*

From the moment both PDKs support at least one simulator in common, merging simulation models from two different PDKs is quite straightforward, as it only requires putting them into separate folders and customizing the cdsinit file. This will ensure that models are correctly loaded when launching Analog Design Environment (ADE). Model names can be post-processed by a simple script to ensure that no duplicates are created.

C. *Pcells*

In order to fit with the new layermap, layout section of pcells code must be modified. FEOL and BEOL layers should be transposed to their new equivalent in the unified PDK as shown on figure 4 below. Moreover, if model name has changed, cdf (Component Description Format) must be customized in order to correctly link the model to the pcell.

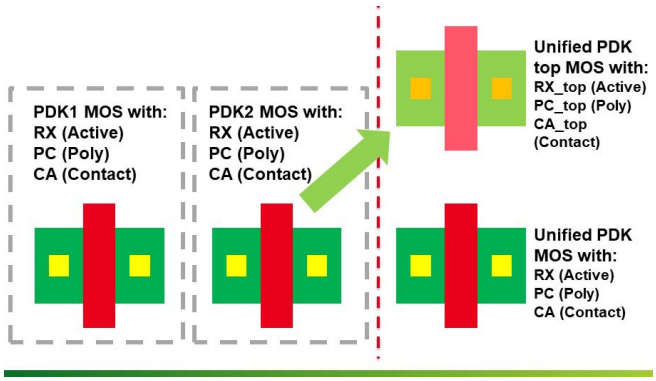

Fig. 4. Pcell layer transposition from PDK1&PDK2 to unified PDK.

D. *Verification Decks (DRC and LVS)*

Verification strategy requires running a complete check for each tier while ensuring that additional checks to verify manufacturability of the inter-tier via are correctly done. From DRC point of view, this means adding specific rules to each tier's deck in order for example to verify correct enclosure of the MIV by top tier first metal layer and bottom tier last metal layer. This is done by adding a post-trigger script, which is executed after bottom tier DRC. This script reconstitutes a PDK2 compatible GDS file by remapping all top tier layers to their original GDS numbers (subtracting the fixed offset value applied before, see section III/A for details). Methodology is shown on figure 5 below.

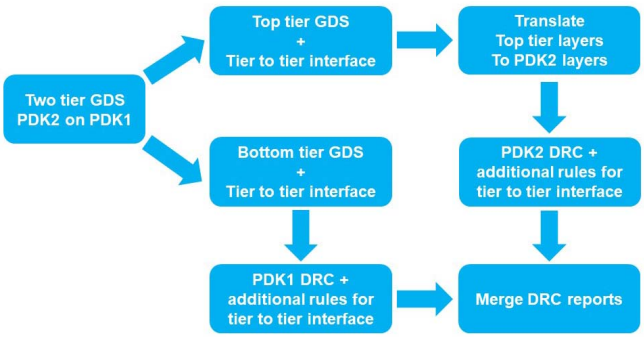

Fig. 5. Methodology for a unified DRC including PDK1 and PDK2 rules with the addition of specific tier to tier interface DRC rules.

From LVS point of view, solution depends on the type of stacking. If PDK2 is different from PDK1 then pcell names are supposed to be different and, contrary to DRC, there is no need to remap layers to their original GDS layers. Therefore a simple merge of Calibre LVS rule files with the addition of the MIV connectivity should allow running a two-tier LVS. However, if PDK2 is similar to PDK1 (logic on logic configuration for instance), then LVS might be trickier to address as it will require to rename every pcell and every derived layer that is occurring in the Calibre LVS rule file.

E. *Analog Parasitic Extraction Deck*

Contrary to digital parasitic extraction for timing-driven place&route which is only extracting BEOL (see section IV for more details), analog parasitic extraction requires a complete description of both FEOL and BEOL for both top and bottom tier. Moreover, description of the tier-to-tier interface (material, physical properties) is as well requested in order to correctly model horizontal and vertical parasitic couplings between MIVs, neighboring BEOL metal layers and implant layers. To our knowledge, no commercial EDA tool is capable of accurately extract parasitics of a multi-tier design. Achievable workaround is to proceed in two phases, extracting each tier separately then merging netlists while adding a resistance to emulate the MIV at each tier-to-tier interface.

IV. Digital Section of the PDK

In this section, we will discuss about digital design environment based on a Synopsys DC compiler flow for synthesis and Cadence Innovus for place and route. Customizing the flow in order to use a single environment for two technologies requires modification of several files such as technology lef, ict file and associated captables and qrctechfile, as well as site definition file, viarule generation file and standard cells libraries.

A. *BEOL Description and Via Generation*

From synthesis to place and route, digital design tools configuration mainly relies on technology lef, viarule lef file and captables/qrctechfile. Technology lef file contains a description of the whole process stack (an alternation of metal and via layers) and all accessible via components in every possible orientation and shape (single, double, horizontal, vertical, etc). Merging technology lef from both PDK requires addition of the inter-tier via which has to be defined between bottom tier last metal layer and top tier first metal layer, paying attention in order to avoid duplicates in metal and via layer description as well as accessible via list.

Viarule lef file contains rules to generate vias but may be superseded by content of the technology lef file. Below is an example of a MIV measuring 100nm by 100nm with center to center spacing of 558nm, metal4 enclosure of 78nm on both x and y directions, top tier metal1 enclosure of 89nm on both x and y directions and connecting bottom tier metal4 to top tier metal1.

```
VIARULE VIA4_RULE GENERATE
      LAYER M4 ;
        ENCLOSURE 0.078 0.078 ;
      LAYER M1_TOP ;
        ENCLOSURE 0.089 0.089 ;
      LAYER CO3D_CC ;
        RECT -0.050 -0.050 0.050 0.050 ;
        SPACING 0.558 BY 0.558 ;
END VIA4_RULE
```

In order to achieve timing-driven routing, place and route tools rely on specific files organized like lookup tables and containing values of R and C parasitics for every combinations of metal and via layers. CapTables are the less precise one and are usually used during first placement steps because they allow faster iterations. Qrctechfiles have the best precision and are therefore used during signoff steps like post-route optimization. To build these files, we need to compile an ict file, which contains process data of all metal and via layers such as dielectric values, thickness, absolute height, resistivity, temperature coefficients, etc. Compilation is done using either Cadence Coyote tool for CapTables or TechGen for QrcTechFiles. Merging PDKs implies that we rebuild CapTables and QrcTechFiles using a new version of the ict file which should contain all BEOL information starting from the lowest metal layer of the bottom tier, going up to the last metal layer of the top tier and including the MIV parameters as well. Complexity to build the ict file is mainly due to height description being absolute, which means that once adding any kind of layer somewhere in the stack, you need to recalculate all the following height values for upper layers. This can be done by hand but requires a lot of time for each iteration or the user can use a templating language like Python Jinja2 to generate a clean and complete merged ict.

B. Standard Cells Libraries

As explained earlier in section II/B, layer consistency is a critical part that must be carefully checked to ensure manufacturability of the 3D circuit. Regarding standard cells libraries, several parts of top tier standard cells libraries must be updated in order to make them compatible with our merged pdk architecture. First part is updating their name (in our case by prefixing them) in order to make sure no duplicates will exist between bottom and top tier standard cells. Duplicates can exist mainly in the case of merging one PDK with itself in order to have the same process technology on both top and bottom tier. However, apart from the duplicates problem, separating top and bottom tier standard cells is mandatory to have two physical implementation versions of the standard cell. One will be implemented using bottom tier FEOL and BEOL, the other using top tier FEOL and BEOL, which cannot have the same layer name as explained in the previous analog section of the PDK.

First part to update is the VHDL or Verilog code, which needs to be modified if standard cells names were changed. Second part is the Cadence library in which we need to change standard cells names as well as update layers used in the layout view and optionally pcells names. Third part is modifying the lef description of standard cells by substituting metal layers with their new equivalents in the unified PDK description. Fourth part is modification of lib files in order to update standard cells names and regenerate db files (which are requested for synthesis). A shell script was developed in order to handle standard cells library migration and cover all the aforementioned topics.

In addition to standard cells libraries, constraints on the MIV (which cannot cross top tier active or poly region) bring the need for specific filler standard cells in order for the place and route tool to place these vias correctly (i.e avoid placing MIVs on top tier standard cells and place them on specific fillers). Moreover, regular top tier standard cells must have an obstruction description to prevent inter-tier vias from being placed over them.

As obstruction can only be set for metal layers, we put an obstruction on the first top tier metal layer which is not supposed to be used for routing as it is already used for intra-cell routing. With this methodology, we are able to route a DRC clean design with timing constraints.

C. Digital Synthesis Flow

For the moment, no 3D synthesis tool is commercially available which means the designer will have to choose a design strategy and handle design split, tier by tier synthesis and merge. This is mainly due to the fact that without any tier information, current synthesis tools won't be able to split the design in two parts (in the case of two tiers) while ensuring proper and proportional distribution of both top and bottom tier standard cells (even if balanced are ratio may not always be the optimal solution as explained in [7]). Therefore, the result would be a synthesized netlist containing only the most suitable standard cells from power, area and speed point of view with respect to user-defined constraints.

D. Digital Place and Route Flow

Digital place and route is suffering from the same drawback as digital synthesis. State of the art contains several workarounds [4][8] in order to achieve physical implementation of a multi-tier digital design, all of them having limitations compared to what could a real 3D place and route engine do. Depending on the design strategy, top tier standard cells lib files may require some updates because of cold-process fabrication which may slightly modify performances of transistors.

V. USE-CASE : EXAMPLE OF RISC-V PROCESSOR

In this section, we will give some more details about design of a RISC-V processor we've just taped-out using ST 28nm FDSOI process technology on the bottom tier and LETI 28nm FDSOI on the top tier. The design was pre-partitioned between tiers in order to take full advantage of a regular 2D place and route run. Memory blocks are implemented on the bottom tier while digital logic is implemented on the top tier. Architecture of the RISC-V processor is detailed below in figure 6. In our case, we used a single BEOL stack distributed over two FEOL tiers, metal1 to metal4 for the bottom tier and metal5 to metal11 for the top tier. This implied that we modify any bottom tier cell having routing over metal4 and particularly I/O pads instances.

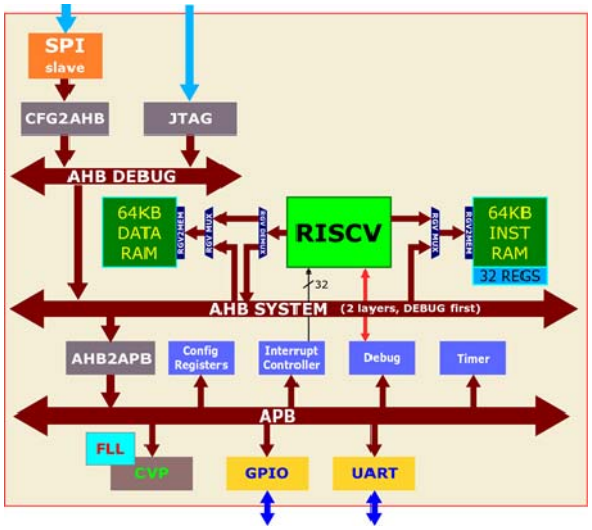

Fig. 6. Design Methodology for the RISC-V test chip.

We created a script that patched all the pads instances by adding a MIV each time it founds a metal4 to metal5 via in order to maintain signal continuity all across BEOL. Figure 7 summarizes the design methodology used for the design of this circuit.

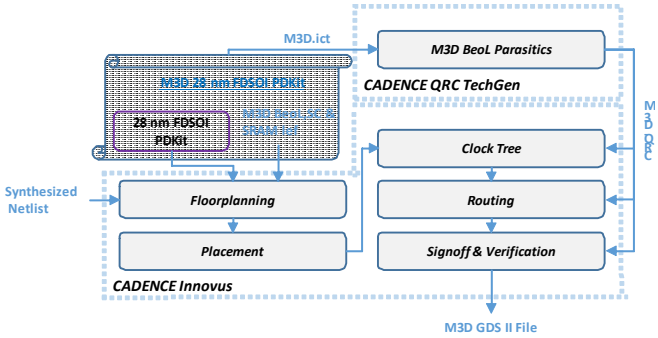

Fig. 7. Design Methodology for the RISC-V test chip.

Signoff verification has been done using an On Chip Variation (OCV) factor of 10%, three process corners for LETI technology: worst, nominal and best with one process corner for the memory blocks in ST 28nm FDSOI: SlowSlow (SS) 0.8V / 125°C. Timing verification was done for several use-cases such as configuration through a SPI, memory pre-load, process boot, GPIO test and a Dhrystone bench for instance.

VI. Conclusion

We created a generic methodology to create a unified design environment by merging PDK information. All possible combinations can be addressed from merging identical PDK1 and PDK2 to merging different PDKs. Our design environment still requests that its source PDKs are installed and available in order to run properly. Challenges and limitations exist because of both EDA tool limitations and encrypted file access. Most of these problems should vanish once a foundry will offer a global solution to 3D sequential process. Among test structures, we taped-out a RISC-V processor based on memory on logic architecture.

Acknowledgment

The authors would like to thank Romain Lemaire for his valuable contribution to the RISC-V demonstrator, especially on the timing validation simulations.

References

[1] Ki-Tae Park et al., “Three-dimensional 128Gb MLC vertical NAND Flash-memory with 24-WL stacked layers and 50MB/s high-speed programming”, 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)

[2] Tsutomu Haruta et al., “4.6 A 1/2.3inch 20Mpixel 3-layer stacked CMOS Image Sensor with DRAM”, 2017 IEEE International Solid-State Circuits Conference (ISSCC)

[3] Samson Melamed et al., “Cool Interconnect: A 1024-bit Wide Bus for Chip-to-Chip Communications in 3-D Integrated Circuits”, IEEE Transactions on Components, Packaging and Manufacturing Technology, Year: 2019, Volume: 9 , Issue: 3, Pages: 525 – 535

[4] Heechun Park et al., “RTL-to-GDS Tool Flow and Design-for-Test Solutions for Monolithic 3D ICs”, 2019 56th ACM/IEEE Design Automation Conference (DAC)

[5] P. Batude et al., “3D sequential integration opportunities and technology optimization”, 2014 IEEE International Interconnect Technology Conference (IITC)

[6] Gérald Cibrario et al., “From 2D to monolithic 3D predictive design platform: An innovative migration methodology for benchmark purpose”, 2016 IEEE International 3D Systems Integration Conference (3DIC), Pages: 1 – 5

[7] Hossam Sarhan, Sebastien Thuries, Olivier Billoint and Fabien Clermidy, “An Unbalanced Area Ratio Study for High Performance Monolithic 3D Integrated Circuits”, 2015 IEEE Computer Society Annual Symposium on VLSI

[8] O. Billoint et al., “A comprehensive study of Monolithic 3D cell on cell design using commercial 2D tool”, 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)