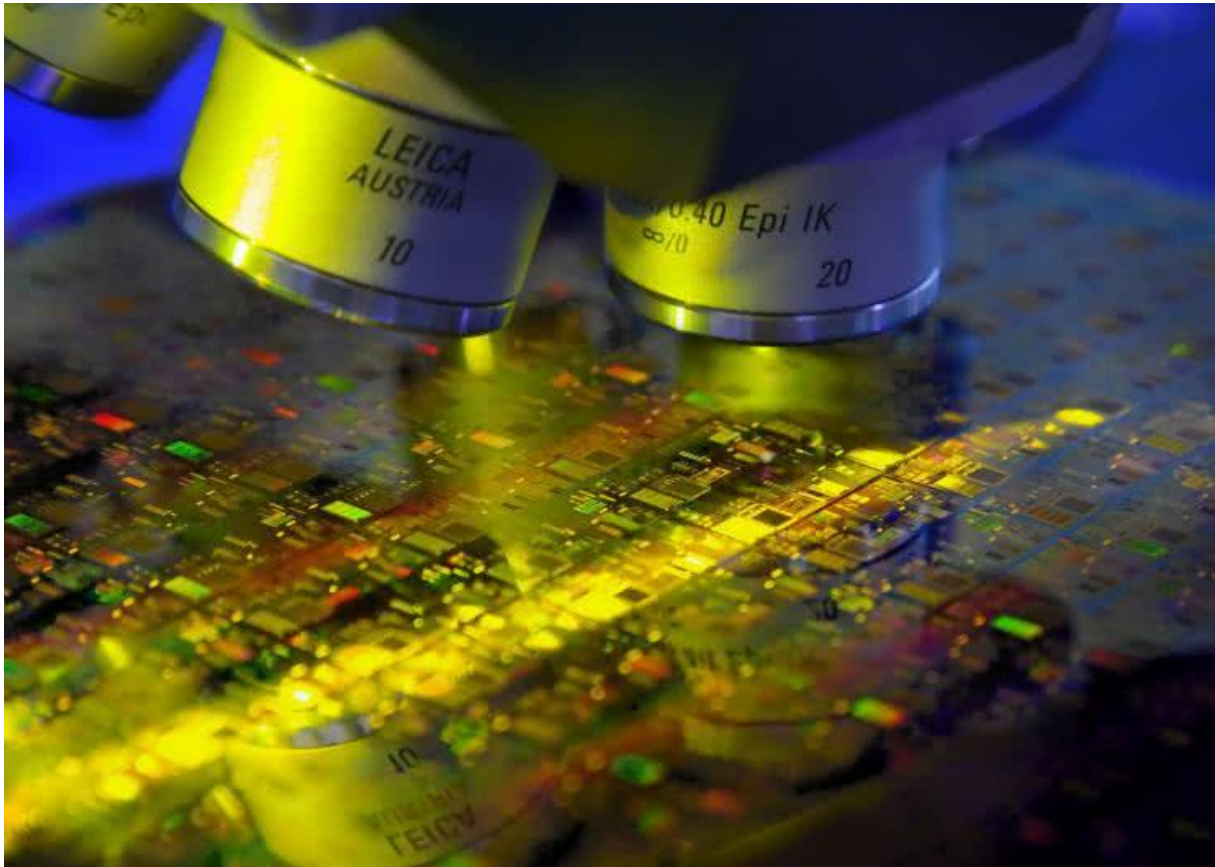


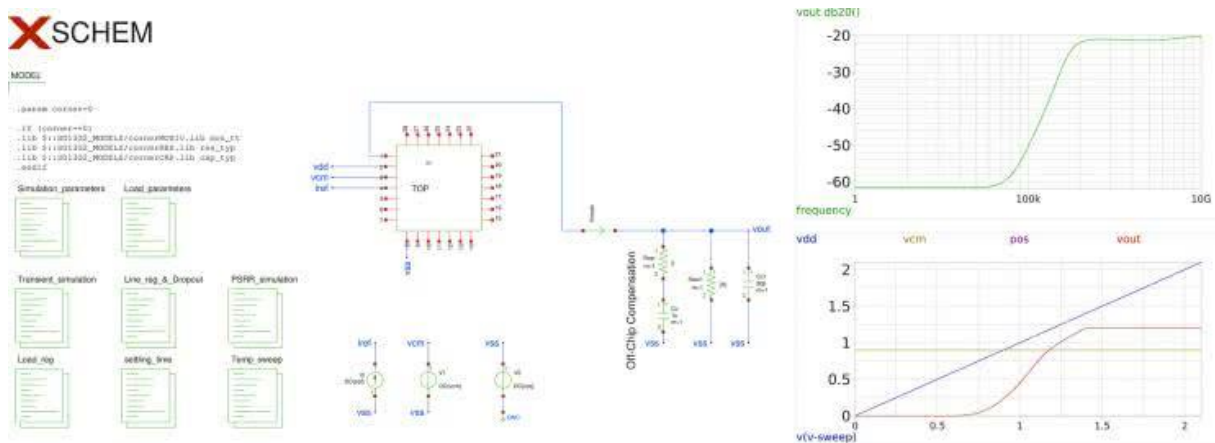
Jak zaprojektować i wyprodukować układ scalony bez nakładów finansowych?

Sobota, 01 Czerwiec 2024



W poprzednim miesiącu obiecaliśmy Czytelnikom zarówno szczegółowy opis rozwiązań implementowanych w instytucie IHP (Leibniz-Institute for High Performance Microelectronics), a służących projektowaniu układów scalonych – jak i prezentację nadchodzących programów Open MPW (Multi Project Wafer), które umożliwiają darmowe opracowanie prototypów układów półprzewodnikowych w technologii SG13G2 BiCMOS 130 nm.

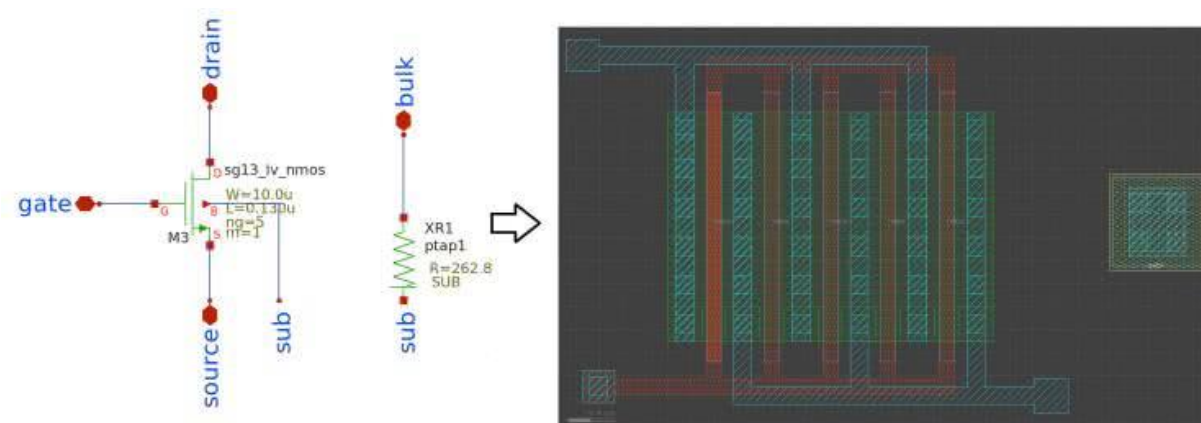
Aby dotrzymać obietnicy, najpierw zajmiemy się narzędziami do projektowania oraz elementami składowymi IHP-Open-PDK [2]. Pierwszym krokiem w kierunku naszego układu ASIC (Application Specific Integrated Circuit) będzie narysowanie jego topologii w edytorze schematów. W domenie open source silicon dostępne są dwa narzędzia do tego celu: Xschem [3] oraz Qucs-S [4], oba wspierane przez IHP. Przykładowy schemat narysowany w edytorze Xschem zamieszczono na rysunku 1.



Rysunek 1. Przykładowy schemat układu LDO narysowany w edytorze Xschem – łącznie z blokami konfiguracji symulacji oraz wizualizacji graficznej symulowanych przebiegów

Kolejnym krokiem będzie przeprowadzenie szeregu symulacji celem weryfikacji działania układu. Tu z pomocą przychodzą dwa narzędzia: ngspice [5] oraz Xyce [6], które są co prawda programami pozbawionymi interfejsu graficznego, niemniej – tak Xschem, jak i Qucs-S – pozwalają nam na skonfigurowanie symulacji oraz wizualizację wyników wewnątrz edytora schematów.

Warto wspomnieć, że wspólnym mianownikiem ww. narzędzi jest tzw. netlista, czyli tekstowy opis tak topologii układu, jak i konfiguracji symulatora. Mając za sobą etap projektowania układu z wykorzystaniem symulacji, możemy przejść do kolejnego kroku, czyli do stworzenia layoutu, który stanowi graficzną reprezentację urządzeń półprzewodnikowych oraz ich połączeń – w tym wypadku na podłożu krzemowym. Krok ten jest odpowiednikiem projektowania płytek PCB z zastosowaniem elementów dyskretnych. Edytorem layoutu, wybranym przez IHP, jest program Klayout [7], powszechnie używany w społeczności open source silicon. Aby ułatwić pracę projektantowi, Klayout pozwala na generowanie urządzeń z wykorzystaniem predefiniowanych szablonów. Każdy szablon posiada szereg zmiennych, które mogą być modyfikowane przez projektanta, tak aby odzwierciedlić wartości parametrów komponentów, zdefiniowanych na etapie symulacji. Zostało to zobrazowane na rysunku 2, ilustrującym pięciobramkowy tranzystor typu NMOS o nazwie sg13_lv_nmos i wymiarach 10,0×0,13 μm . W związku z tym, że proces SG13G2 realizowany jest na podłożu typu P, odpowiedni kontakt z podłożem zapewnia element ptap1, modelowany w symulacji z użyciem rezystora.



Rysunek 2. Przykład schematu opracowanego w Xschem oraz layoutu wykonanego z użyciem Klayout

Wydawałoby się, że wygenerowanie elementów, ich rozmieszczenie oraz połączenie kończy pracę nad układem – nic bardziej mylnego! W przypadku opracowywania płytek PCB odpowiedzialnością projektanta jest przestrzeganie reguł projektowych, związanych z możliwościami procesu technologicznego oraz dostarczanych przez fabrykę, która płytki fizycznie produkuje. Podobnie jest w domenie półprzewodników: poszczególne procesy, w tym proces SG13G2, narzucają w niej reguły projektowania, których należy bezwzględnie przestrzegać. Można znaleźć je w dokumencie SG13G2_os_layout_rules.pdf w repozytorium GitHub IHP-Open-PDK. Z pomocą w tym trudnym zadaniu projektowania layoutu zgodnie z wieloma regułami przychodzi nam edytor Klayout – jest on w stanie zweryfikować poprawność naszego projektu z użyciem reguł DRC (Design Rule Check), dostarczonych i zaimplementowanych przez autorów IHP-Open-PDK.

Kolejnym krokiem, nowym w porównaniu do projektowania PCB, jest krok weryfikacyjny, nazywany LVS (Layout Versus Schematic). Ma on za zadanie porównanie naszego schematu (netlisty w formacie tekstowym) z netlistą wyekstrahowaną z layoutu. Proces ekstrakcji obejmuje ekstrakcję elementów (oraz ich parametrów) i połączeń między nimi. Celem ww. kroku jest osiągnięcie spójności między schematem i layoutem. Podobnie jak w przypadku DRC, weryfikację LVS można przeprowadzić za pomocą programu Klayout, w którym zaimplementowane zostały reguły ekstrakcji oraz porównywania obu reprezentacji układu (schematu vs. layoutu).

Krokiem specyficznym dla układów elektronicznych pracujących powyżej 10 GHz (fale milimetrowe) są symulacje rozkładu pola elektromagnetycznego, dla wybranych części obwodu elektronicznego, celem ich charakteryzacji za pomocą elementów takich jak linie długie. W IHP trwają działania nad opracowaniem ww. metod z wykorzystaniem narzędzia openEMS [8] oraz jego integracji z opisanym wcześniej oprogramowaniem (Qucs-S, ngspice, Xyce, Klayout). OpenEMS może być użyty w kolejnym, charakterystycznym dla układów

wysokich częstotliwości kroku, związanym z wartościami pasożytniczymi oporności, pojemności, indukcyjności (własnej i sprzężonej), zwanym PEX (Parasitics EXtraction). Wartości wygenerowane na etapie PEX wiążą się głównie z właściwościami elektrycznymi połączeń oraz geometrii układu opracowanej na etapie rysowania layoutu. W kroku tym możliwe jest wyekstrahowanie ww. wartości pasożytniczych i uwzględnienie ich w symulacji (post layout simulation), w celu zwiększenia dokładności otrzymanych wyników, które stają się tym samym bardziej rzeczywiste. Ostatni krok w procesie projektowania stanowi wygenerowanie pliku w popularnym formacie GDS2, który jest akceptowany przez większość fabryk półprzewodników.

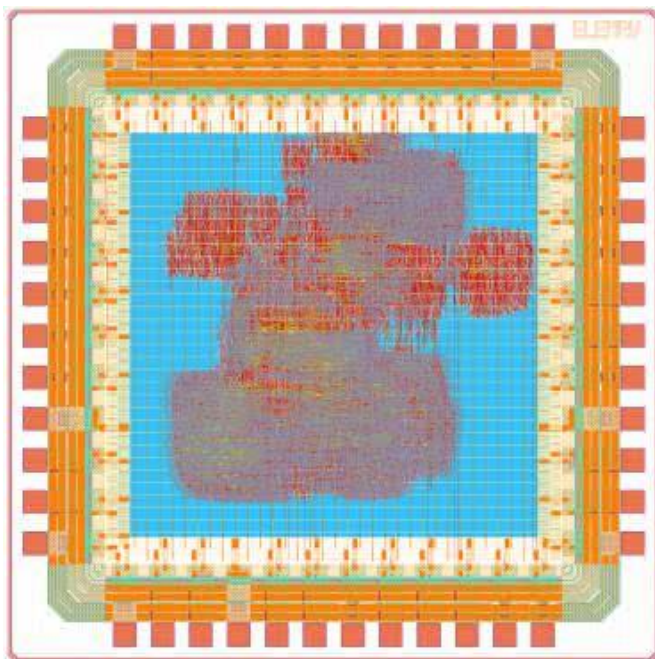
Podsumowując, projektowanie jest procesem iteracyjnym i polega na wykonaniu sekwencji kroków z użyciem odpowiednich narzędzi. W wielu aspektach opracowywanie układów scalonych przypomina tworzenie układów elektronicznych z zastosowaniem elementów dyskretnych. W przypadku technologii półprzewodnikowych projektowanie sprowadza się głównie do określenia topologii układu oraz rozmiarów/formy użytych elementów półprzewodnikowych. Bardzo ważną ogólną zasadą pozostaje wykonywanie weryfikacji tam, gdzie to tylko możliwe, w związku z potrzebami: zapewnienia poprawnego działania układu, detekcji oraz eliminacji błędów.

Aby zapoznać Czytelnika z możliwościami rozwoju układów scalonych w obrębie technologii IHP SG13G2, przechodzimy do opisu elementów składowych IHP-Open-PDK. Sam PDK zachowuje kompatybilność z komercyjnym procesem SG13G2 dostarczany przez IHP. Jego cechą charakterystyczną jest obecność tranzystorów bipolarnych HBT (w trzech wersjach) do zastosowań w bardzo wysokim paśmie częstotliwości rzędu setek gigaherców. Z komponentów aktywnych w PDK znajdziemy także komplementarne tranzystory polowe w wariantach niskiego (1,2 V) i wysokiego (3,3 V) napięcia.

Projektant ma ponadto do dyspozycji diody oraz struktury zabezpieczające ESD. W obszarze podłoża krzemowego możemy wykonać również 3 typy rezystorów o rezystancji powierzchniowej od kilku omów do kilku kiloomów. Połączenia w procesie SG13G2 są realizowane za pośrednictwem siedmiu warstw aluminiowej metalizacji (komercyjna wersja procesu dostarcza opcje wykonania metalizacji na bazie miedzi). Warto nadmienić, że ostatnie dwie warstwy metalu są grube (osiągają 3 μm) w porównaniu do cienkich, liczących 0,5 μm warstw o numerach 1...5, co z kolei ułatwia prowadzenie ścieżek, którymi płyną duże prądy. W obrębie warstw metalizacji istnieje możliwość wykonania kondensatora MiM (Metal Insulator Metal), induktorów oraz transformatorów. Poprzez dostępność wyżej opisanych elementów PDK rozumiemy istnienie bibliotek projektowych oraz niezbędnych danych, koniecznych, aby

zaprojektować i przetestować układ z użyciem narzędzi, takich jak np. Xschem, ngspice i Klayout. Zgłębiając strukturę PDK: znajdziemy biblioteki symboli edytorów schematów; definicje modeli SPICE łącznie z danymi odzwierciedlającymi rozrzut parametrów w procesie produkcyjnym, które są przydatne w analizie typu Monte Carlo; szablony layoutów elementów półprzewodnikowych; zestaw reguł DRC i LVS. Dodatkowo PDK dostarcza bibliotek niezbędnych w procesie projektowania układów cyfrowych z zastosowaniem narzędzia OpenROAD-flow-scripts (ORFS), dostępnego na platformie GitHub [9].

Znajdziemy tam biblioteki standardowe (głównie bramki logiczne, przerzutniki), układy IO, pamięci SRAM. Każda z bibliotek zawiera modele: (1) funkcjonalny – w postaci kodu verilog, (2) czasowy – w postaci plików w formacie liberty, (3) funkcjonalny w domenie analogowej – w postaci plików SPICE, (4) uproszczonej geometrii – w postaci plików LEF, (5) layoutu – w postaci pliku GDS2. Narzędzie ORFS otrzymuje oficjalne wsparcie ze strony IHP, co pozwala na projektowanie układów cyfrowych, które następnie mogą zostać wyprodukowane w Instytucie. Przykład takiego układu, procesora RISCV, zaprezentowano na rysunku 3, a jego projekt dostępny jest na GitHubie [10].



Rysunek 3. Layout procesora RISCV wykonanego z użyciem elementów dostępnych w ramach IHP-Open-PDK

Zgodnie z informacjami zaprezentowanymi w poprzednim artykule, sam proces rozwijania układów elektronicznych na krzemie w domenie open source może zostać rozszerzony o możliwości opracowania prototypów poprzez dostęp do zdolności produkcyjnych wybranych fabryk lub linii pilotażowych w ramach programów Open MPW. Instytut IHP udostępnił harmonogram organizacji programów Open MPW zamieszczony w tabeli 1.

Harmonogram organizacji programów Open MPW oferowanych przez IHP w ramach projektu FMD-QNC							
Tape Out date	10 May 2024	11 Nov 2024	22 Nov 2024	01 Mar 2025	09 May 2025	18 Jul 2025	15 Sep 2025
Technology	SG13G2	SG13CMOS	SG13G2	SG13G2	SG13G2	SG13G2	SG13CMOS
Area [mm²]	10	220	20	140	30	30	220

W ramach OpenMPW IHP oferuje 2 procesy technologiczne – SG13G2 oraz SG13CMOS – między którymi jedyną różnicę, z punktu widzenia projektanta, stanowi dostępność tranzystorów HBT. Obecne są one tylko w procesie SG13G2 i pozwalają na projektowanie układów w paśmie setek GHz.

Pomimo że szczegółowe informacje na temat samego procesu oraz sposobu zgłoszenia udziału w programie OpenMPW można znaleźć na stronie [11], kilka aspektów wymaga szerszego wyjaśnienia. Najważniejszym spośród wszystkich wymagań opisanych na ww. stronie jest gotowość do upublicznienia swojego projektu na zasadach open source (licencja Apache 2.0). Pozwoli to na opracowanie biblioteki układów elektronicznych zaprojektowanych przez partycypantów programu, która w przyszłości stanie się bazą do projektowania bardziej skomplikowanych konstrukcji. Aby zagwarantować jakość ww. elementów biblioteki, układy elektroniczne wyprodukowane w kolejnych edycjach programów OpenMPW będą mogły podlegać testowaniu nie tylko przez projektanta, ale także każdego, kto będzie chciał takowych testów dokonać. Warto podkreślić, że projektant nie otrzyma swojego układu na własność, jednak będzie mógł wypożyczyć go z biblioteki układów (IHP-OpenSampleStoreHouse) – tak jak każda inna osoba zainteresowana testowaniem i dostarczeniem dodatkowych informacji na temat działania układu. W ramach współpracy badawczej IHP może także udostępnić swoje zdolności pomiarowe podmiotom zainteresowanym na prowadzenie badań naukowych.

Uogólniony harmonogram programu OpenMPW został pokazany na rysunku 4. Mimo że szczegółowe informacje na temat każdej edycji będą publikowane na bieżąco, należy już dziś wiedzieć, że sam proces produkcji trwa w przybliżeniu 6 miesięcy, a okres ewaluacji planowany jest na maksymalnie 2 lata.



Rysunek 4. Uogólniony harmonogram programu OpenMPW oferowanego przez IHP w latach 2024...2025

Aby z sukcesem wziąć udział w najbliższym programie OpenMPW, warto jak najszybciej zainstalować niezbędne narzędzia, sklonować nasze repozytorium IHP-Open-PDK oraz zacząć przygodę z open source silicon. W razie jakichkolwiek pytań zapraszamy do kontaktu z nami drogą mailową lub na kanale Slack [12] #ihp-sg13g2. Zapraszamy także do śledzenia publicznie dostępnych materiałów z konferencji, takich jak FSIC [13], ORConf i Latch-Up [14].

dr Krzysztof Herman, dr Anna Sojka-Piotrowska

herman@ihp-microelectronics.com, sojka@ihp-microelectronics.com

IHP GmbH – Leibniz Institute for High Performance Microelectronics, Frankfurt (Oder),
Niemcy

<https://www.ihp-microelectronics.com>

Ważne linki:

1. <https://www.ihp-microelectronics.com/>
2. <https://github.com/IHP-GmbH/IHP-Open-PDK>
3. <https://xschem.sourceforge.io/>
4. https://github.com/ra3xdh/qucs_s
5. <https://git.code.sf.net/p/ngspice/ngspice>
6. <https://github.com/Xyce/Xyce>
7. <https://www.klayout.de/>
8. <https://github.com/thliebig/openEMS-Project>
9. <https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts>
10. <https://github.com/IHP-GmbH/IHP-Open-DesignLib/tree/main/ElemRV>
11. <https://ihp-open-ip.readthedocs.io/>
12. <https://open-source-silicon.slack.com>
13. https://wiki.f-si.org/index.php?title=Main_Page
14. <https://fossi-foundation.org/>