Tape-Out Course: Silicon in a Semester

Dan Fritchman, *Member, IEEE* Aviral Pandey, *Member, IEEE* Kris Pister, *Member, IEEE* Ali Niknejad, *Member, IEEE* Borivoje Nikolić, *Member, IEEE*

For UC Berkeley's pandemic-disrupted 2021 spring semester, eighteen students - four PhD candidates, six MS, and eight undergraduates - signed up for a "special topics" course based on little more than a terse description:

In this class, we will design and send out for fabrication a system on chip (SoC) intended for internet-of-things applications (IoT). The chip will contain a RISC-V microprocessor, a radio transceiver, and a baseband signal processor, and will be designed in a 28nm CMOS process (And we really mean it!).

Less than four months later the fully remote student design team had produced and submitted such an SoC for fabrication. The student-designed SoC, nicknamed "OsciBear" by its designers, includes a 32-bit RISC-V processor core, hardware AES encryption and decryption acceleration, and a mixed signal IEEE 802.15-compatible Bluetooth Low Energy transceiver. It consumes 1.0 mm sq in TSMC's 28nm HPC technology.

This article chronicles the effort of the student design team, and that of their instructors to provide course-based practical tape-out experience to largely first time student-designers, working in a large and diverse team to produce a complex system-on-chip.

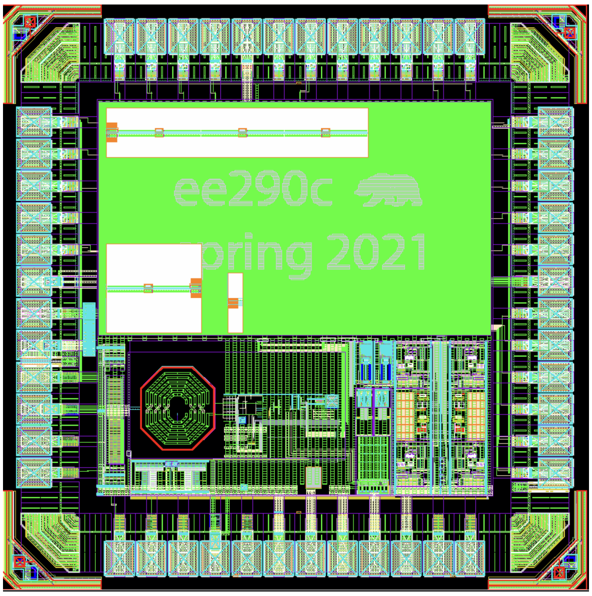

Fig. 1. OsciBear SoC Layout

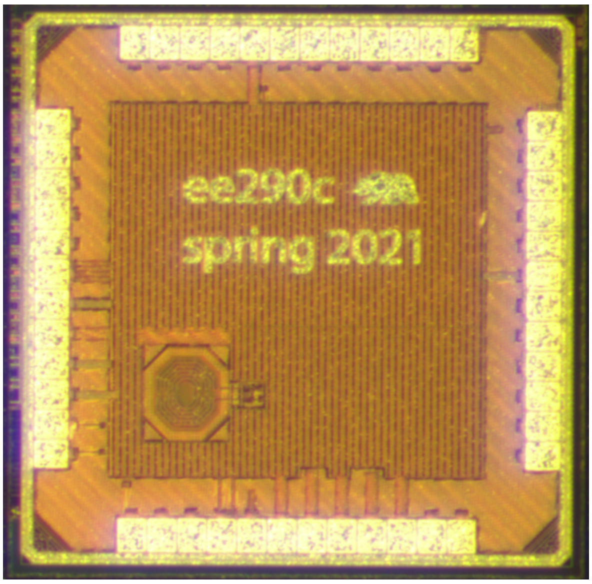

Fig. 2. OsciBear Die Micrograph

I. SoC Design Course

Titled *28nm SoC for IoT*, the offering was Berkeley's fourth recent effort to provide a course-based, hands on tape-out experience. The course's fairly atypical goals include:

- Exposing younger, typically first time, student-designers to the realities of a complete silicon design process and tape-out. Most of the students' time is spent on topics lightly covered by traditional circuit courses, such as digital back end design (logic synthesis, place and route layout), custom analog & RF layout, and the technical and interpersonal demands of working in a large team.
- Demonstrating the capacity for such a small group to produce a complex IC on a constrained schedule.
- Simulating a large team environment more representative of commercial IC design than typical academic projects, especially those undertaken by BS and MS candidates.

Most IC design students do not meet an opportunity to participate in a full chip design and tape-out until the masters or more often doctoral level. Even more rare is the experience of working together in a large and diverse team with a variety of roles and backgrounds. In recent years industry demand for these skills and experiences has dramatically outpaced academia's ability to train graduate students at these levels. One remedy would be dramatically increasing the quantities of these advanced degrees. Our effort instead examines whether more students can pursue them much earlier in their careers. Such practical IC courses were once common, following the introduction of renowned editions by Carver and Mead. In subsequent decades industry and academic practice diverged sufficiently to render them impractical. We find that our field has shifted such that they are both practical and valuable once again.

Our course's initial iteration in 2017, detailed in [1],

Date of publication March 28, 2022.

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720 USA.

attempted the design of a similar mixed-signal SoC targeted for micro-robotics research. It succeeded in designing and fabricating an RF transceiver, but was unable to integrate its analog and digital subsystems. More ambitious attempts at more feature rich processor and transceiver designs followed in 2018 and 2019 and were unable to reach completion within the sixteen week semester. The Spring 2021 iteration detailed in this article was the first to produce a complete mixed-signal SoC.

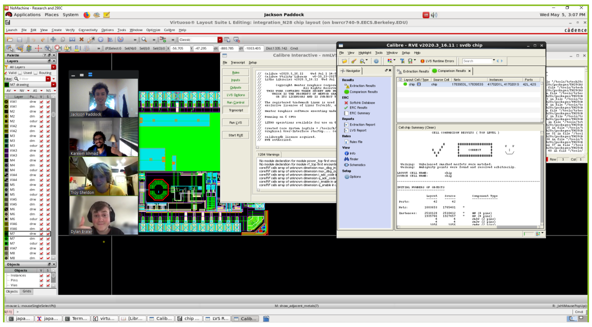

Fig. 3. Student-Designers Completing Chip-Level Verification

II. Course & Project Organization

The tape-out course is atypically organized, with no traditional assignments, no traditional exams, and no solutions manual. Instead students completed one large "course project", with one large team: designing the SoC. No two students' "coursework", i.e. design contributions, were alike.

The course also features very little traditional lecturing. Roughly 75% of lecture meetings were dedicated to students presenting design meeting style updates. The remaining 25% of lecture time was largely dedicated to practical topics, such as best practices in custom layout, hierarchical design flow tools, PCB and package design, and post-silicon verification. These lectures were spread among the three primary faculty and two graduate assistants, plus several guest lecturers from Berkeley and industry partners. The students presented two lengthy design reviews with industry partners, one roughly midway through the course and one shortly after its completion.

The effort is also atypical for a design team - particularly in that upon design start, the team members and leaders knew very little about the backgrounds and interests of one another. The course's first step was conducting a student survey including three primary questions:

- Their background courses in *digital* circuits,
- Their background courses in *analog and/or RF* circuits,
- Their interests and goals for joining the course and working on the SoC.

Even at such an early stage in their educations and careers, we observed the students had already largely broken off sub-specializations: only three of the eighteen had taken courses in both digital and analog circuits. Prerequisite digital IC courses generally feature HDL design in Verilog targeting FPGAs. Few students held prior experience with a typical VLSI flow, or the realities of producing fabrication ready layout. Analog and RF students entered having studied circuit analysis and performed small design projects. Few had been exposed to design in modern technologies, or to custom layout of large and elaborate circuits. Students were offered either of two labs to guide setting up relevant EDA tools. A digital VLSI flow lab demonstrated a build process from Berkeley's hardware description language, Chisel [2], to Verilog and through industry standard back end tools to layout. An analog lab focused on the practicalities of cross corner SPICE simulation and layout design. Later tutorials added depth of layout best practices and efficient editing.

III. Design Process

The OsciBear SoC's content was chosen primarily by its designers. After surveying backgrounds and breaking the group into teams, the instructors provided two required blocks for inclusion in the SoC: the RISC-V processor and RF transceiver. The students had a wide space to choose additional peripherals and features. The initial block diagram requirements and suggested extensions provided by the instructors are show in Figure 4.

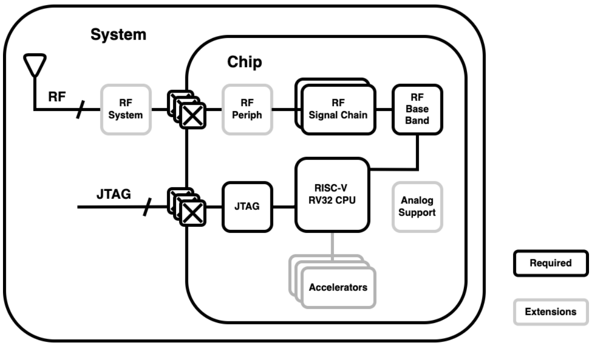

Fig. 4. Initial Block Diagram Requirements

The students elected to produce an AES encryption accelerator, an RF digital baseband, on-die RF LO generation, and on-die power management. Alternatives included acceleration for edge machine learning and a digital phase locked loop for SoC clock generation. The RF transceiver scope omits several components which would desirably be included on-die in a commercial SoC, notably including a power amplifier and transmit-receive switch. These elements were left to system-level integration for sake of design time. A block diagram of the OsciBear SoC is shown in Figure 5.

A. Processor / Compute Complex

The SoC's digital subsystem includes three primary components: the RISC-V CPU, AES accelerator, and RF baseband processor. Each of the three took fairly different routes to design, particularly with respect to reuse of past work.

The past decade of Berkeley EECS research has produced a broad array of both IC design content and related *design productivity* software, much of which the OsciBear SoC depended directly upon. This includes the RISC-V [3] instruction set, the Chisel [2] hardware description library, the Rocket Chip generator [4] and associated TileLink bus, the Hammer [5] EDA flow and back end framework, and the ChipYard [6] integration framework, incorporating all of the above. OsciBear is an example circuit produced by a *design generator* program using this "Berkeley design suite".

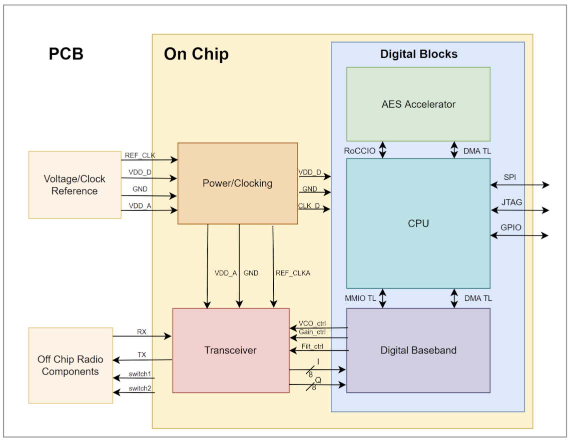

Fig. 5. OsciBear SoC Block Diagram

Rocket and ChipYard's generators are configured through an elaborate set of Scala language configuration classes, such as the excerpted OsciBear SoC configuration shown below.

```
class EE290CBLEConfig extends Config(
  // ...
  new baseband.WithBLEBasebandModem ++
  new aes.WithAESAccel ++
  new WithBSel ++
  new WithNGPIOs(3) ++
  new chipyard.config.WithSPIFlash ++
  new EE290Core ++                 // single tiny rocket-core
  new WithEE290CBootROM ++         // use our bootrom
  new WithNEntryUART(32, 32) ++    // add a UART
  new freechips.rocketchip.subsystem.WithJtagDTM ++
  new freechips.rocketchip.system.BaseConfig)
```

Identifiable lines within `EE290CBLEConfig` dictate much of the SoC's core content: its Rocket core, inclusion of the baseband processor and AES accelerator, its GPIOs, SPI flash, JTAG debug transfer module (DTM), and its UART. Note the excerpt from `EE290CBLEConfig` includes content designed by the student team (`EE290Core`, `WithAESAccel`), and more reused from the Rocket and ChipYard projects (`WithSPIFlash`, `BaseConfig`). These configuration classes serve as primary input to a generator program which negotiates bus widths, address spaces, and many more tedious design details.

The student sub-team tasked with designing the compute complex - the processor, its features, and its primary interactions with peripheral modules - initially focused on design studies using this configuration API to optimize for the target embedded use case. In later stages they team then became leads in the software integration and back end design effort. While primarily designed to operate with a 20MHz clock to support the default Bluetooth DSP sample rates, the digital subsystem closed timing at 50MHz, allowing for further CPU performance exploration for non-Bluetooth use cases. It consumes 16.9mW from a 900mV supply at 50MHz, and 11.8mW at 20MHz. The digital layout is shown in Figure 6.

B. AES Encryption Acceleration

Student-designers of the AES encryption/ decryption accelerator module adopted a different approach. Their design centrally uses an open source SystemVerilog-based AES implementation designed and published by SecWorks. This core logic interacts with the remainder of the OsciBear Soc via a Chisel-designed controller. The controller accepts requests via custom instructions defined via the Rocket Chip Coprocessor Interface (RoCC), and transfers input and output data through DMA. A block diagram of the accelerator is show in Figure 8.

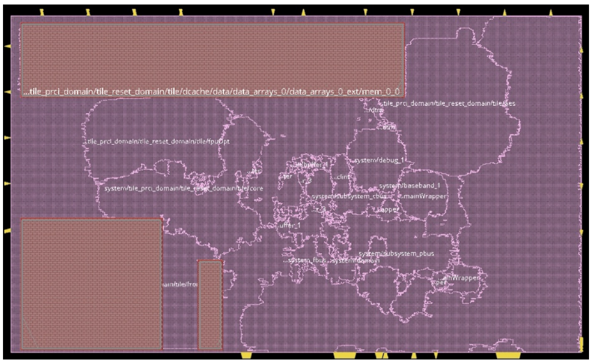

Fig. 6. Digital Subsystems Layout

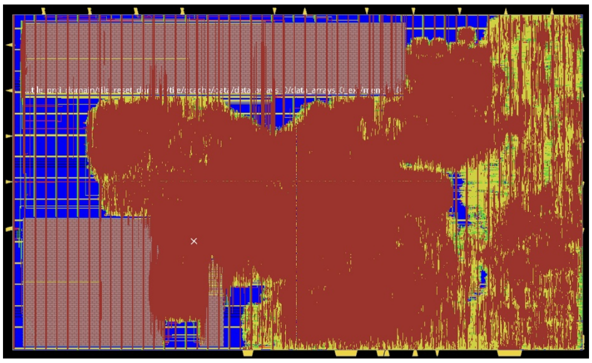

Fig. 7. Digital Subsystems Layout

A key effort of the AES accelerator's design was comprehensive design verification. The designers' simultaneous master's thesis work focused on design verification; which paired well with incorporating, controlling, and verifying an open source IP core. The designers co-developed a CHISEL based verification framework, and used said framework to verify both the AES accerator from SEC works, and their controller and RoCC interface.

C. Bluetooth Low Energy Transceiver and Baseband Processor

The OsciBear SoC includes a mixed signal IEEE 802.15-compatible Bluetooth Low Energy transceiver, which includes both an analog front-end and a digital baseband. The baseband was created from scratch by its student-designers and includes the BLE modem, modulation and demodulation DSP. It further contains link layer components such as packet recovery, cyclic redundancy checks, and data whitening. A block diagram of the baseband processor is pictured in Figure 9.

The BTLE analog transceiver is shown in Figure 9, with its transmit pad at top-left and receive pad at bottom-left. An on-die PLL shown in Figure 11 generates its local

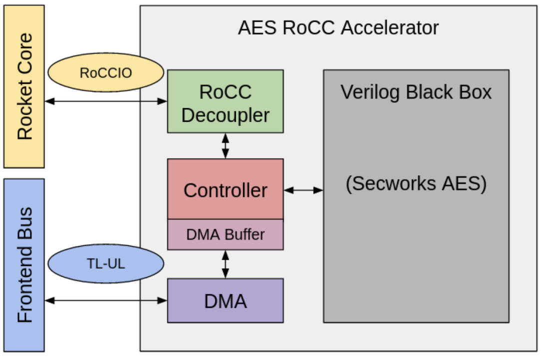

Fig. 8. AES Encryption Accelerator Block Diagram

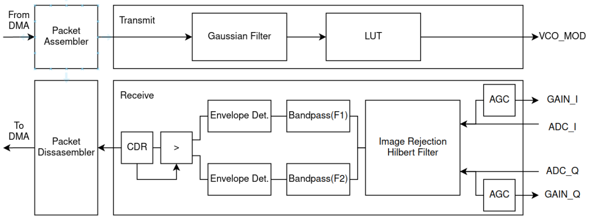

Fig. 9. RF Baseband Processor Block Diagram

oscillator from a 2MHz reference. The BTLE transmitter uses a direct modulation architecture, using the *GFSK Tune* input to the PLL to directly frequency-shift the LC oscillator pictured in Figure 12). The BTLE receiver uses a low-IF architecture, featuring the passive mixer shown in Figure 13), programmable gain amplifier, and bandpass filters. All were designed from scratch, as were the VCO, PLL, and a pre-PA. Schedule constraints forced the SoC to omit several features which would desirably be integrated in commercial BLE transceivers, including a power amplifier, matching networks, and a transmit-receive switch. These components are instead integrated at the PCB level. The receiver ADCs and peripheral test circuits were supplied by the course staff.

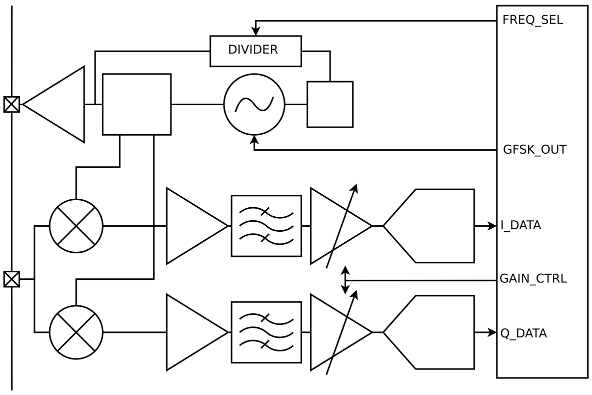

Fig. 10. BlueTooth Low-Energy Transceiver Block Diagram

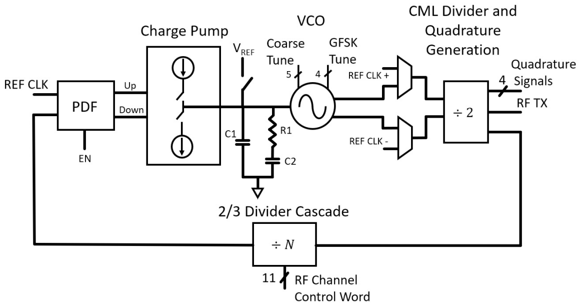

Fig. 11. RF LO Generation PLL

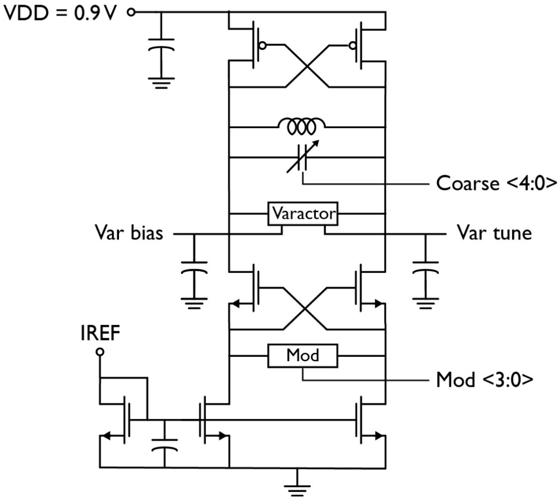

Fig. 12. RF LO VCO

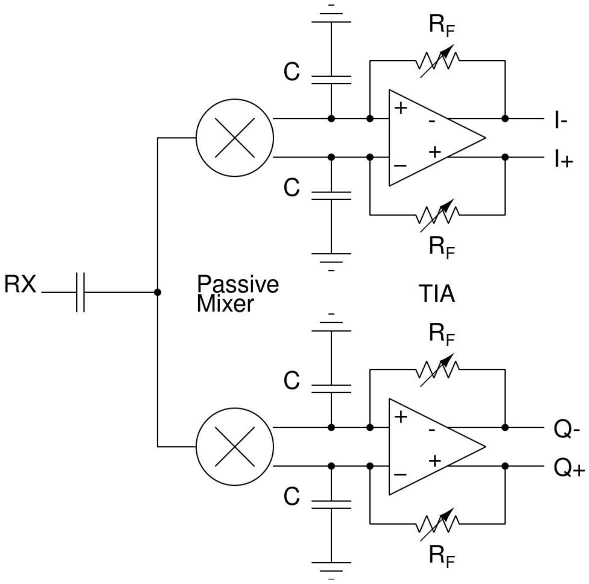

Fig. 13. RF Mixer

In contrast to its digital designers, the course's analog designers found little to reuse. Process technology disclosure requirements typically prevent publication of analog designs on resources such as GitHub; these designs must therefore be walled off per institution. The TSMC 28HPC technology had not been used by Berkeley researchers prior to the course, rendering no such in-house library available. (A similar 28nm technology had been used extensively by Berkeley researchers and became the basis for several peripheral

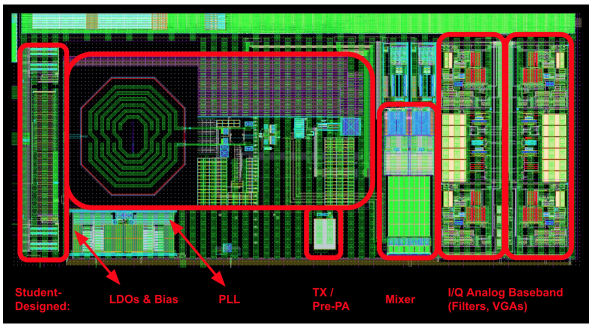

Fig. 14. RF & Analog Sub-Systems Layout

circuits.)

Recent Berkeley research has also produced a suite of productivity software for process-portable analog IC design, centrally including Berkeley Analog Generator (BAG) [7] framework. While BAG was deployed by the course's instructors for the RF transceiver's ADC, the remaining student-designed RF circuits were designed in an industry-standard environment, using Cadence's Virtuoso design suite and Mentor Graphics's physical verification suite (Calibre LVS and DRC).

D. Power Management

The OsciBear SoC is powered by a single cell compatible supply of between 1.2 and 1.5V. On-die linear voltage regulators then convert this VBATT into two 900mV supplies, one each for the digital and analog subsystems. Each identical regulator uses a the topology shown in Figure 15. The regulator feedback is output compensated, allowing arbitrarily high decoupling capacitance, of particular benefit to the digital subsystem. Each regulator output is fed to a chip level pad, allowing for a large off chip compensation and decoupling capacitance, and allowing for separate analog and digital power measurements with the on-die regulators are disabled. With a 10mA load the regulator achieves greater than 47dB of power supply rejection.

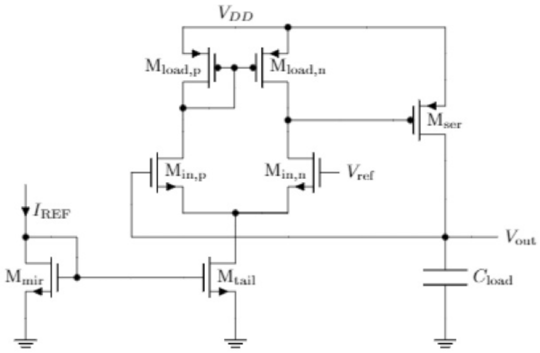

Fig. 15. On-Die Supply Voltage Regulator

E. Verification

In a typical industrial SoC design process, a large portion of engineering effort is dedicated to verification. Academic research ICs tend to be completed by smaller teams with more modest validation efforts. The OsciBear effort landed somewhere in between. No student-designer was tasked full time with verification duty, but all were tasked with contributing to the effort. Helpfully, the design productivity suite includes infrastructure for compiling and executing to a target Rocket configuration. This proved especially valuable for chip level simulations executing target software. The AES and baseband subsystems featured more targeted verification, aided by their designers' codesigned Chisel verification frameworks.

The RF transceiver began from spreadsheet based modeling of a link budget and performance targets. Coupled with a target architecture, it then deployed Verilog-A models of each major transceiver subcomponent (mixer, VCO, etc.) before committing to detailed schematic and layout design. The parameters of these simplified analog models then served as requirements for the performance of each component, which could be readily checked in simulation and review.

Integration testing between analog and digital subsystems commonly remains a challenging industry task. The relevant timescales and execution models for processors executing software and transistor level RF circuits differ dramatically, leaving full co-simulation over relevant execution times impractical. The student-designers instead developed Verilog behavioral models of the crucial analog components, particularly capturing their relevant interactions with the digital subsystems. These models allowed for select RTL domain simulation with the baseband and remaining digital components.

IV. Limitations

Berkeley's spring 2021 tape-out course was an unmitigated success, described by one faculty member as a "minor miracle". Should more students, both at Berkeley and at other institutions, take similar courses? While we find this mode of hands on, true to life instruction highly effective, we must note several factors make it unlikely to scale to all IC design students.

The course included three faculty members, comprising expertise in RF circuits, digital circuits, and robotics, respectively, and two graduate teaching assistants. This student to teacher ratio is about as good as we can imagine finding, particularly at large public institutions.

The mix of students also proved virtually ideal, both across experience and interest. The course roster included a roughly even mix of grad and undergrad (4 PhD, 6 MS, 8 BS). The design experience of the graduate cohort proved particularly invaluable for tasks steeped in industry specialist terms and tools, such as designing and analyzing phase noise of the LO VCO, and generating adequate constraints for digital back end flows. More experienced students served informal leadership and mentoring roles to their younger counterparts. We note that while omnipresent in research and industry settings, this form of teamwork is rare in course based projects, and therefore to students only exposed to course based projects. Roughly half of students expressed interest in analog and/or RF circuit design, well paired with the effort required to complete the BTLE transceiver.

The course chip also heavily relied upon, and likely only succeeded because of, the corpus of Berkeley design productivity software. While designing a simple (or even complex) processor core can often be used as an academic course project, the litany of peripherals, buses, and software support that make up the Rocket and ChipYard projects could not. In many cases the students had direct or indirect access to these works' primary authors, many of whom provided invaluable support.

Last, while the course proves that design and tape-out of a mixed-signal SoC can be done in one (whirlwind) semester, the chip's life does not end at tape-out. Fabrication, PCB design, and testing necessarily extend beyond the duration of a university semester. This work began with the design of the custom test PCB shown in Figure 16, designed by student Jeffrey Ni. As of this writing, post-silicon work remains in progress, led by a combination of its remaining student-designers and students in the Spring 2022 offering. We find that the lab based bring up experience offers student-designers valuable perspective, particularly with regard to the utility of debug-targeted features and of thorough and well documented verification. We intend for future offerings of the course to carry on this pipeline, in which students perform a combination of post-silicon work on recently designed chips and design of their own, incorporating their lab borne insights.

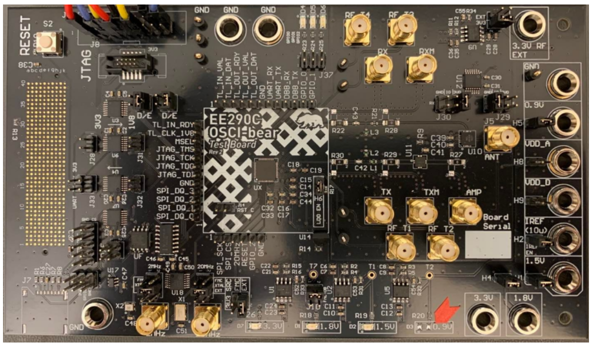

Fig. 16. Assembled test PCB. OsciBear SoC at center.

We also note several potential factors which *were not* limiting factors, and we expect would not be at similar institutions. Notably: the costs to fabricate the custom silicon through an academic multi-project wafer program totaled less than $20k, facilitated by our partners at Muse Semiconductor. Associated costs of fabricating circuit boards were even lower. These "direct" material costs were likely less than many courses featuring a lab component, whether for circuit design or the physical sciences. Second: commercial EDA software, which often comes at high cost to commercial designers, is commonly licensed at low or no cost to academic institutions. Our student designers used the same compute and EDA infrastructure as the Berkeley Wireless Research Center (BWRC)'s research designs. Both of these factors - access to and cost of silicon, and cost of EDA software - would likely have been prohibitive to the same group of eighteen designers in a commercial environment (i.e. a theoretical "OsciBear, Inc."). Academic courses such as this offer a unique opportunity to provide access to these resources.

V. Student Feedback

Shortly after the tape-out and semester's conclusion, the student-designers were asked to complete a survey consisting of three short questions:

- What were the most and least enjoyable aspects of this course?
- Overall, what did you think of this style of course relative to more typical ones?
- Has it made you more or less interested in doing more of this as a job or field of research?

While students' technical contributions varied widely, their thoughts on these topics clustered into a few common themes.

A. Dislike: Quality of Tools and Their Support

Many students noted their frustrations with the complex design software stack. These frustrations targeted both commercial EDA and home grown research generated software in similar amounts. Examples included:

Re: *Which were the least enjoyable aspects of this course?*

- "The least thing I liked about this class was that the tools were not setup correctly which made me spend a lot of time figuring out solutions."
- "Picking up the ChipYard tool in a short period of time. It was really nice to learn ChipYard, but if it had a more official and dedicated channels, I think it would be great and less bothering to ChipYard developers!""

We empathize with these frustrations. Berkeley EECS students study a combination of topics typically subdivided into EE and CS departments, and generally entered our course with some level of programming skill. These skills are typically based in popular, open languages (e.g. Python, Java, C) and libraries (e.g. TensorFlow, PyTorch) for which immense resources are publicly available. When problems arise their solutions are often only a web search away. Commercial EDA tools, research borne IC design software, and silicon process technologies typically lack these amenities. Finding local experts in each proved essential for the student-designers' success.

B. Dislike: Over-Communicating, Over-Meeting

The course format included very little traditional lecturing, and was near entirely dedicated to student-presented updates in a design meeting style format. Many students cited the volume of these updates as a complaint.

Re: *Which were the least enjoyable aspects of this course?*

- "(The) amount of logistics that went into everything and the fact that basically half of the time the lecture was going to be a little irrelevant if it focused on a different team working on something orthogonal to what you're working on."
- "(The) constant check-ins during lecture."

Again we empathize. Academic course projects typically include far more individual contribution time and far less communication time than the OsciBear SoC effort. These design meetings more patterned an industrial design environment. (We note the total time dedicated to these sessions was typically between 3 and 5 hours per week, a paltry amount for many industry designers.)

Moreover, a large and diverse project such as an eighteen designer mixed-signal SoC has many widely varied technical sub-projects. Student-designers took varied levels of interest in subsystems outside their own. Students were recommended, although not required, to attend each other's design updates and provide feedback and questions, particularly on subsystems which interacted with their own designs.

Further, the focus of these sessions changes substantially over a project's timespan. Early stages focus heavily on architectural design and planning. Students focused on these facets were heavily involved, while students focused on later stage content featured less prominently. Later stages more heavily feature physical design, logical and physical verification. These roles then essentially exchanged. In an industrial setting, these varying focuses would often be performed by specialist teams and pipelined among projects. Academic research-designers, in contrast, specialize less, and contribute more outside their core interest areas. Student-designers were encouraged to do the latter, with the acknowledgement that they could only reasonably dive into so many technical areas.

C. Overall Impressions & Future Interest

Re: "*Overall, what did you think of this style of course relative to more typical ones?*"

- "I think the time requirement is a lot higher. Perhaps being real clear about this at the beginning would help a lot of students."
- "I really enjoyed the freedom we were afforded along with the openness of the staff (professors & great TAs!). It was great that we were able to explore the design process but had a lot of support and expertise from the teaching staff... overall this was one of the best classes I've taken at Berkeley and the stuff I learned will stay with me forever during my career."
- "Best course I've taken at Berkeley!"

Re: "*Has it made you more or less interested in doing more of this as a job or field of research?*":

- "I think this has made more interested in doing this as a job but I also recognize how much effort goes into it."
- "More interested! The design process was very fun in my opinion, as we had the opportunity to go through each stage of the design process and see our design come to life: design -> implementation -> unit verification -> integration w/ SoC -> SoC verification. I am interested in designing bigger and more complex designs!"

As noted in their overall impressions, we believe students worked unusually hard at the tape-out course, dedicating substantially more of their time relative to more typical courses. Nonetheless their impressions of the process were overwhelmingly positive. Of the eleven survey respondents, nine reported that the course had made them *more* likely to pursue further work or research in the area. The remaining two reported their interests stayed "about the same".

Enthusiasm for the tape-out-based course clearly reached their student peers. For the ongoing spring 2022 edition, the course's enrollment nearly tripled to over fifty students.

VI. Acknowledgements

Designing a complex SoC takes a village. Doing so in a university semester with primarily first time designers takes an even larger one. The authors offer particular thanks to:

- Our guest lecturers, Daniel Grubb and James Dunn of UC Berkeley and Simone Gambini.
- The patient and attentive researcher-authors of all UC Berkeley's home grown IP and design software, particularly including Abraham Gonzalez, Harrison Liew, and Zhaokai Liu.
- Muse Semiconductor for designing and coordinating the academic multi-project wafer.
- TSMC for providing educational access to the 28nm HPC technology.
- Financial support and design review from our industry partners, especially Jared Zerbe and Ramesh Abhari.

But most of all the authors thank and congratulate our eighteen incredibly talented and enthusiastic students: Nayiri Krzysztofowicz, Josh Alexander, Cheng Cao, Troy Sheldon, Kareem Ahmad, Dylan Brater, Jackson Paddock, Felicia Guo, Daniel Fan, Alex Moreno, Shreesha Sreedhara, Kerry Yu, Sherwin Afshar, Eric Wu, Anson Tsai, Ryan Lund, Griffin Prechter, and Jeffrey Ni.

References

[1] D. C. Burnett, B. Kilberg, R. Zoll, O. Khan, and K. S. J. Pister, “Tapeout class: Taking students from schematic to silicon in one semester,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.

[2] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek, and K. Asanović, “Chisel: constructing hardware in a scala embedded language,” in *DAC Design automation conference 2012*, pp. 1212–1221, IEEE, 2012.

[3] K. Asanović and D. A. Patterson, “Instruction sets should be free: The case for risc-v,” *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146*, 2014.

[4] K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, *et al.*, “The rocket chip generator,” *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17*, vol. 4, 2016.

[5] E. Wang, A. Izraelevitz, C. Schmidt, B. Nikolic, E. Alon, and J. Bachrach, “Hammer: Enabling reusable physical design,” in *Workshop on Open-Source EDA Technology (WOSET)*, 2018.

[6] A. Amid, D. Biancolin, A. Gonzalez, D. Grubb, S. Karandikar, H. Liew, A. Magyar, H. Mao, A. Ou, N. Pemberton, *et al.*, “Chipyard: Integrated design, simulation, and implementation framework for custom socs,” *IEEE Micro*, vol. 40, no. 4, pp. 10–21, 2020.

[7] J. Crossley, A. Puggelli, H.-P. Le, B. Yang, R. Nancollas, K. Jung, L. Kong, N. Narevsky, Y. Lu, N. Sutardja, *et al.*, “Bag: A designer-oriented integrated framework for the development of ams circuit generators,” in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 74–81, IEEE, 2013.