

**Филиппов Егор Олегович**

студент

**Бескодаров Егор Александрович**

студент

*Научный руководитель*

**Марченко Анна Константиновна**

старший преподаватель

ФГБОУ ВО «МИРЭА – Российский технологический университет»

г. Москва

## **ОБУЧЕНИЕ ПРОФЕССИОНАЛЬНОМУ АНГЛИЙСКОМУ ЯЗЫКУ ЧЕРЕЗ КОНТЕКСТ МИКРОЭЛЕКТРОНИКИ: ИНЖЕНЕРНЫЕ СИМУЛЯТОРЫ КАК ЛИНГВОДИДАКТИЧЕСКАЯ СРЕДА**

***Аннотация:** в статье рассматривается вопрос применения инженерных симуляторов для более эффективного освоения студентами технических направлений профессионального английского языка. Основная идея состоит в том, что язык легче запоминается не через отдельные списки терминов, а в процессе реальной работы: при чтении англоязычного интерфейса, разборе HDL-кода, поиске ошибок компиляции и обращении к технической документации. На примере микроэлектроники рассматриваются три варианта такой работы: параметрическая симуляция, проверка HDL-модуля и составление отчёта об ошибке.*

***Ключевые слова:** профессиональный английский язык, микроэлектроника, инженерные симуляторы, CLIL, LTspice, HDL, Icarus Verilog, GTKWave, лингводидактическая среда.*

*Введение.*

Современная индустрия полупроводников организована глобально: проектирование, производство и верификация чипов распределены между командами разных стран, а английский язык фактически выступает единым рабочим стандартом. На нём написаны спецификации, документация инструментов и техни-

ческие задания. Традиционный подход к языковой подготовке инженеров – ESP (English for Specific Purposes) – обычно строится на аудиторной работе с адаптированными текстами и терминологическими списками. Этот формат важен: на занятиях студент получает концептуальную базу, разбирает принципы работы устройств и формирует системное понимание предмета. Однако после аудитории язык часто не закрепляется через действие: термин, усвоенный на паре, не встречается затем в живом профессиональном контексте и быстро забывается. В статье предлагается решение этой проблемы: использовать инженерные симуляторы как среду самостоятельной домашней работы, где профессиональный английский закрепляется через применение. Цель работы – обосновать эту модель и описать сценарии её реализации.

## *Раздел 1. Языковая экосистема инженера в области ИС-дизайна.*

### *1.1. Специфика профессионального дискурса.*

Язык, с которым ежедневно работает инженер в области ИС-дизайна, отличается от английского стандартных аудиторных курсов. Его можно представить как трёхуровневую систему. Первый уровень – языки описания аппаратуры (HDL): Verilog, VHDL, SystemVerilog. Их синтаксис построен на английской лексике: *always*, *if*, *begin*, *end*, *assign*, *input*, *output* – это осмысленные слова, понимание которых помогает понять логику схемы. Второй уровень – интерфейсы EDA-инструментов: меню, команды, сообщения об ошибках и предупреждения симулятора создают постоянный поток профессиональной лексики. Третий уровень – техническая документация: даташиты, спецификации IP-блоков, руководства по интеграции. Эти жанры имеют устойчивую структуру и специфическую риторику. Аудиторная работа даёт представление об этих уровнях, но не обеспечивает полного погружения; для этого нужна самостоятельная работа в среде, близкой к реальному профессиональному контексту.

### *1.2. Психологический барьер и его природа.*

Студенты технических специальностей нередко испытывают трудности с «гуманитарным» английским: чтением связных текстов, развёрнутыми высказываниями и грамматикой вне контекста. При этом англоязычная среда инже-

нерного симулятора часто воспринимается легче, потому что инженерное мышление опирается на структуру, логику и быструю обратную связь. На занятии обратная связь опосредована преподавателем, а результат правильного употребления термина остаётся абстрактным. Симулятор, используемый дома, действует иначе: неверно понятый параметр даёт неправильный график, а неточно прочитанная инструкция приводит к ошибке компиляции. Студент самостоятельно взаимодействует с англоязычной средой, и это формирует практическую языковую компетентность. Такой механизм соответствует принципам CLIL (Content and Language Integrated Learning): язык устойчиво усваивается тогда, когда служит инструментом решения предметной задачи.

*Раздел 2. Концептуальная модель: симулятор как среда для самостоятельной работы.*

### *2.1. Теоретическая рамка: CLIL в инженерном контексте.*

Методология CLIL возникла как ответ на ограниченность изолированного языкового обучения. Её центральный тезис состоит в том, что язык усваивается эффективнее, когда становится средством достижения лингвистической цели. В предлагаемой модели аудиторное занятие выполняет функцию введения: студент получает концептуальную основу на родном языке или в смешанном формате. Домашнее задание в симуляторе закрепляет материал: именно здесь CLIL работает наиболее полно, поскольку студент взаимодействует с англоязычной средой самостоятельно, без перевода и посредника. Важно, что профессиональный инженерный софт изначально англоязычен, поэтому он не требует искусственного конструирования учебной среды.

### *2.2. Механизм усвоения: визуальный образ – действие – термин.*

Ключевое преимущество симуляторов перед текстовыми материалами состоит в том, что они соединяют визуальный образ, действие и языковой знак. Например, на паре студент узнаёт о модуляции длины канала в MOSFET-транзисторе. Дома в LTspice он изменяет параметр channel length modulation и видит, как меняются выходные характеристики. Одновременно он наблюдает результат на графике, выполняет действие и читает английский термин в его

функциональном значении. Такая фиксация отличается от повторного чтения определения из учебника. Аналогично в цифровой среде студент, отлаживающий HDL-модуль, встречает *rising edge*, *propagation delay*, *undefined state* не как слова для запоминания, а как описания поведения схемы.

### *2.3. Инструментальная база.*

Практическая реализация модели не требует доступа к дорогостоящему коммерческому программному обеспечению. На учебном уровне достаточно открытых инструментов: LTspice для аналоговых схем и SPICE-симуляции, Logisim Evolution для цифровой логики, Icarus Verilog совместно с GTKWave для симуляции и визуализации HDL-кода. Все три инструмента полностью англоязычны, бесплатны и имеют развитую документацию – то есть воспроизводят языковую среду профессионального уровня при нулевых затратах. По мере освоения студенты переходят к коммерческим EDA-платформам – Synopsys, Cadence – доступным через университетские лицензии. Этот переход сам по себе является языковым событием: терминология, усвоенная дома в LTspice, обнаруживается в интерфейсе профессионального инструмента, создавая эффект узнавания и снижая психологический барьер.

### *Раздел 3. Три кейса реализации.*

Предлагаемый подход не заменяет традиционную аудиторную работу. На занятиях студент знакомится с устройством элементов схемотехники, разбирает тексты и выполняет задания под руководством преподавателя; этот этап формирует концептуальную базу. Симуляторные задания выполняют другую функцию: выносятся в формат самостоятельной домашней работы и закрепляют материал. Термин, разобранный на паре, студент затем встречает в контексте симуляции, а действие и визуальный результат переводят знание из кратковременной памяти в долговременную. Ниже описаны три сценария такой работы с разной языковой сложностью.

#### *Кейс 1. Параметрическая симуляция.*

*Инструмент:* LTspice.

*Языковая нагрузка:* чтение и интерпретация англоязычного технического задания, работа с SPICE-параметрами, анализ сообщений симулятора.

После аудиторного занятия, на котором разбирается принцип работы биполярного транзистора, студент получает домашнее задание исключительно на английском: настроить схему усилителя, последовательно изменяя параметры модели – forward current gain (hFE), saturation current (Is), base transit time (TF) – и фиксируя изменения на графике выходных характеристик. Задание намеренно не сопровождается переводом: незнакомый термин должен быть понят через контекст симуляции. Языковое событие здесь – момент, когда студент самостоятельно соотносит слово saturation в тексте задания с визуальным насыщением кривой на графике. Это соотнесение, совершённое без посредника, формирует значительно более устойчивую семантическую связь, чем аудиторное объяснение. Итог домашней работы – короткий письменный комментарий на английском: что изменилось, почему, какой параметр оказался критическим.

*Кейс 2. Верификация HDL-модуля.*

*Инструмент:* Icarus Verilog + GTKWave.

*Языковая нагрузка:* чтение и написание HDL-кода, интерпретация сообщений компилятора, работа с assertions.

Кейс логично следует за аудиторным занятием по цифровой логике или основам HDL. Дома студент получает готовый модуль с намеренно внесёнными ошибками и задачу: написать тестбенч на SystemVerilog, верифицировать поведение модуля и локализовать дефекты. Самостоятельный характер работы здесь принципиален: столкнувшись с сообщением компилятора implicit declaration или timing violation без возможности немедленно спросить преподавателя, студент вынужден интерпретировать английский текст ошибки как единственный источник информации – именно так это происходит в реальной профессиональной практике. Синтаксис тестбенча при этом выступает одновременно инженерным и языковым упражнением: конструкции initial begin, if (output!==(expected), \$display («Error: expected %b, got %b»)) требуют понимания как логики схемы, так и английских условных структур.

*Кейс 3. Отчет об ошибке по результатам симуляции.*

*Инструмент:* любой симулятор из предыдущих кейсов.

*Языковая нагрузка:* написание структурированного отчёта об ошибке на английском языке.

Этот кейс завершает цикл и является наиболее сложным по языковой нагрузке, поэтому подходит как итоговое домашнее задание. Студент оформляет найденный дефект в виде bug report по стандартной структуре: Title, Environment, Steps to Reproduce, Expected Behavior, Actual Behavior, Severity. Каждый раздел тренирует отдельный навык: заголовок – краткую номинацию проблемы, Steps to Reproduce – цепочку императивных действий, Expected / Actual Behavior – противопоставление двух состояний системы. Важно, что этот жанр реально используется в индустрии. Готовый bug report можно разобрать на следующем занятии, замыкая цикл самостоятельной и аудиторной работы.

*Анализ по итогам кейсов.*

Три кейса образуют последовательную дидактическую лестницу. Первый работает на уровне рецепции: студент читает англоязычное задание и понимает термины через результат симуляции. Языковая нагрузка здесь умеренная: ошибка в понимании даёт неожиданный график, который сам указывает на недопонимание. Второй кейс переходит к продукции: студент читает и пишет на английском, а сообщения компилятора становятся обратной связью. Третий выводит на уровень профессиональной коммуникации: bug report является полноценным жанровым текстом с требованиями к структуре, краткости и однозначности.

Важно, что все три уровня опираются на аудиторную базу и не существуют без неё. Студент, не разобравший на паре принцип работы транзистора, не сможет осмысленно выполнить первый кейс – симуляция превратится в механическое нажатие кнопок без понимания. Именно это разграничение делает модель методологически устойчивой: аудиторная работа отвечает за понимание, самостоятельная работа в симуляторе – за закрепление и перевод знания в навык.

*Заключение.*

Настоящая статья предлагает переосмыслить роль инженерных симуляторов в подготовке специалистов по микроэлектронике. Традиционная модель языковой подготовки строится по принципу последовательности: сначала язык, потом профессия. Предлагаемая модель работает по принципу параллельности: профессиональные задачи, решаемые в англоязычной среде симулятора, становятся инструментом языкового закрепления – не вместо аудиторной работы, а в дополнение к ней.

Практическая значимость подхода определяется несколькими факторами. Во-первых, он не требует пересмотра учебных программ: домашние задания в симуляторе органично встраиваются в существующий формат курса. Во-вторых, он масштабируем: открытые инструменты – LTspice, Logisim, Icarus Verilog – доступны любому студенту вне зависимости от технической оснащённости вуза. В-третьих, он готовит к реальной профессиональной среде точнее, чем адаптированные учебные тексты: терминология, усвоенная через симуляцию, – это именно тот язык, который студент встретит в команде верификации или в интерфейсе коммерческого EDA-инструмента.

Перспективы развития подхода связаны с двумя направлениями. Первое – методическое: разработка структурированных наборов домашних заданий, согласованных с аудиторными темами и выстроенных по нарастающей языковой сложности. Второе – технологическое: появление ИИ-ассистентов внутри инженерного программного обеспечения открывает возможность создания двуязычных обучающих надстроек для САПР, которые могут адаптировать сложность англоязычного интерфейса под уровень конкретного студента.

В более широком смысле предложенная модель отражает сдвиг в понимании того, что значит учить профессиональный язык. Не язык для инженеров, а инженерия как способ учить язык.

### *References*

1. Coyle D., Hood P., Marsh D. CLIL: Content and Language Integrated Learning. Cambridge: Cambridge University Press, 2010.

2. Dudley-Evans T., St John M. J. Developments in English for Specific Purposes: A Multi-Disciplinary Approach. Cambridge: Cambridge University Press, 1998.
3. Basturkmen H. Developing Courses in English for Specific Purposes. London: Palgrave Macmillan, 2010.
4. IEEE Standard for SystemVerilog – Unified Hardware Design, Specification, and Verification Language. IEEE Std 1800–2017. New York: IEEE, 2018.
5. Linear Technology / Analog Devices. LTspice XVII Documentation. Analog Devices, 2024.