

# **Samsung Moscow SoC team**

# Про System-on-Chip team

Команда System-on-Chip:

- существует 10+ лет
- разрабатывает компиляторы и связанные проекты (статический и динамический анализ и другой тулинг)
- используемые технологии: in-house, Clang/LLVM, GCC

# Что классно

- State-of-the-art технологии
- Возможность "потрогать" результат руками
- Дружелюбный коллектив
- Есть у кого учиться

# Достижения (1)

- Разработка двух поколений компилятора для нейросетевого ускорителя (в high-end-телефонах Samsung Galaxy)
  - [Ускоряем нейросеть на уровне железа: интервью с разработчиком компиляторов](#)
- Поддержка фронтенда OpenACC в компиляторе GCC
- Разработка компилятора для DSP-процессоров
  - [DSP-процессоры: назначение и особенности](#)

# Достижения (2)

- Поддержка межмодульного анализа в Clang Static Analyzer
  - [2016 LLVM Developers' Meeting: A. Sidorin "Summary-based inter-unit analysis for CSA"](#)
- Поддержка новых фич в AddressSanitizer LLVM и GCC (continue-after-error, порт на ARM, etc.)
  - [GCC devmeeting 2017: Applying GNU GCC Address Sanitizer to whole Linux distribution](#)
  - [GCC devmeeting 2019: Annotating std::string with AddressSanitizer](#)

# Достижения (3)

- NDA :(

# Планы

- Компиляторы для application-specific ускорителей
  - ["The Future of Computing: Domain-Specific Accelerators"](#)  
[William Dally](#)
- Верификация (может быть...)

# Кого мы ищем

NPU-компилятор:

- NPU Compiler Developer for Exynos AI Accelerator (<https://hh.ru/vacancy/42341825>)

High-performance computing:

- [GPU performance engineer](#)
- можно без опыта если знаете и умеете в CUDA и MPI

Разработчик DSP-компиляторов:



# Что надо будет делать (1)

- Кодить на C++, Python, shell (редко asm)
- Отлаживаться: понять почему *опять* упал Jenkins, копаться в логах, читать ассемблер, разбираться в спеках по микроархитектуре девайса, etc.
- Ревуить чужой код
- Общаться (S. Korea, China, US, Israel)

# Что надо будет делать (2)

- Читать новые (и старые) статьи по предметной области
- Исследовать и править open-source код (LLVM, GCC, ROCm, etc.)
  - можно коммитить в open-source :)
  - если есть время :(

# Зачем мне это?

- Делать современный тулинг в высокопрофессиональной международной команде
- Знакомиться с cutting-edge технологиями компиляции и HW-ускорения

# Задать вопросы ???

TG @the\_real\_yugr