2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

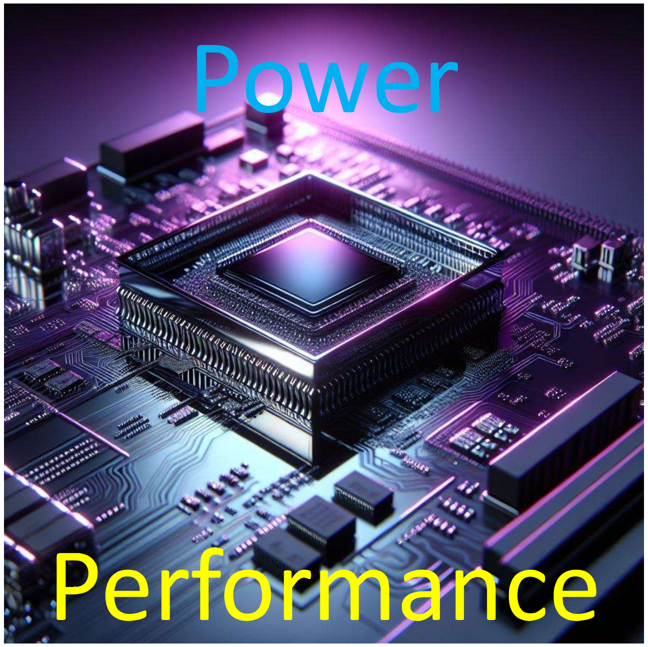
Moving Application-level Power Optimization to Pre-silicon with Advanced Hybrid Emulation and Power Exploration Technologies

Malte Doerper, Leonard Drucker, Ritesh Goel

Agenda for Today

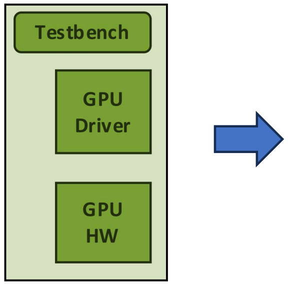
- Application-level software – should a DV engineer care
- Where do I find the software to do my new DV job
- New technology enablers for Application-level Hybrid Emulation
- Application-level Hybrid Emulation examples
- Outlook

What Design Objectives to serve, and why ?

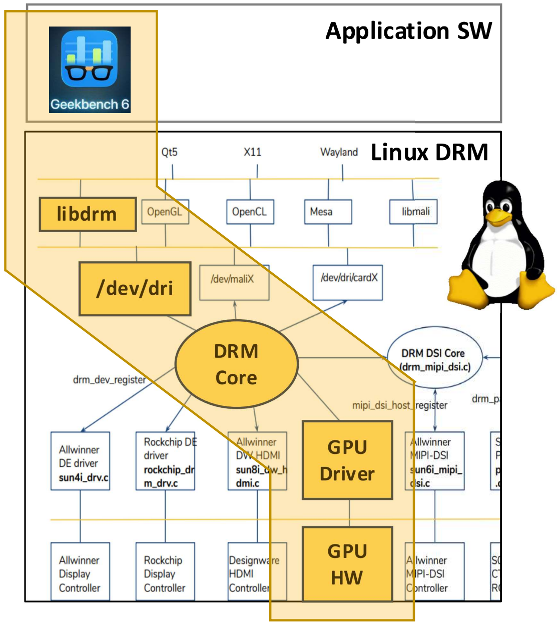


- User Experience
- Market Competitiveness
- Efficiency and Reliability
- Thermal Management
- Battery Life
- Regulatory Compliance
- Cost Reduction

Product’s success defined by application performance



DV with Driver SW

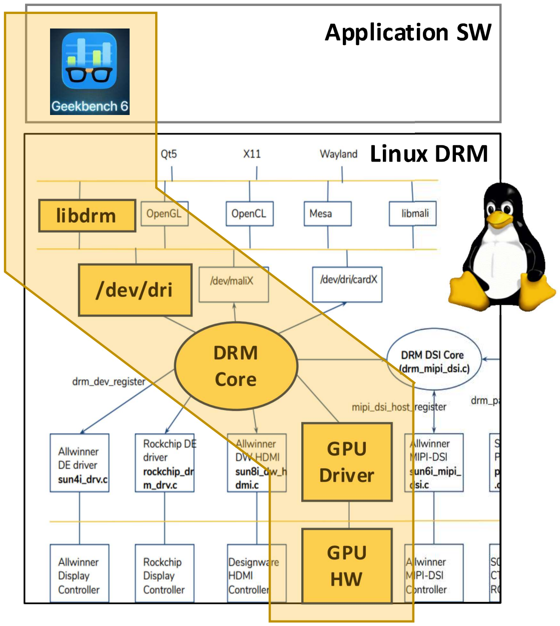


Application SW
Geekbench 6
Linux DRM
Qt5
X11
Wayland
libdrm
OpenGL
OpenCL
Mesa
libmali
/dev/dri
/dev/maliX
/dev/dri/cardX
DRM Core
DRM DSI Core (drm_mipi_dsi.c)
drm_dev_register
mipi_dsi_host_register
drm_p
Allwinner DE driver sun4i_drv.c
Rockchip DE driver rockchip_drm_drv.c
Allwinner DW HDMI sun8i_dw_hdmi.c
GPU Driver
Allwinner MIPI-DSI sun6i_mipi_dsi.c
Allwinner Display Controller
Rockchip Display Controller
Designware HDMI Controller
GPU HW
Allwinner MIPI-DSI Controller

DV with Application Workload

- Software that impacts semiconductor performance starts at the application layer
- Majority of verification happens at driver layer with testbench workloads
- Such an approach is insufficient to validate product success
- DV engineers needs to work with SW teams and use entire SW stack

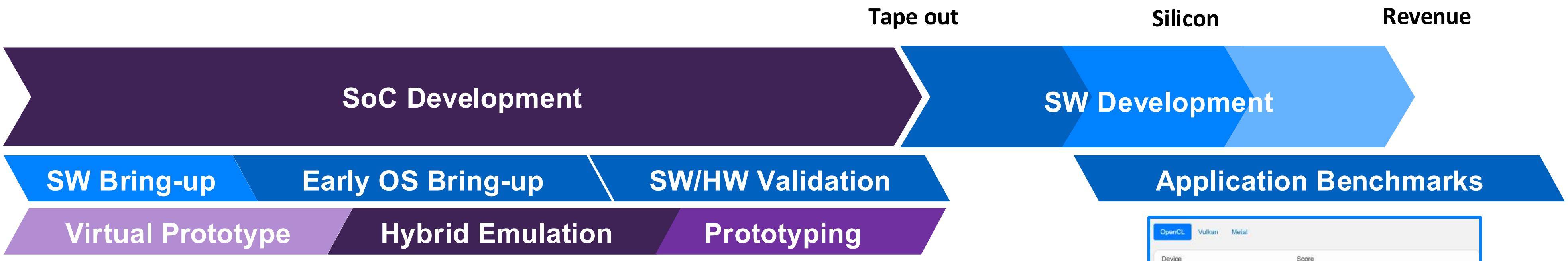
Application Validation Requires Entire Software Stack

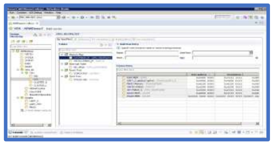

Application SW
Geekbench 6
Linux DRM
Qt5
X11
Wayland
libdrm
OpenGL
OpenCL
Mesa
libmali
/dev/dri
/dev/maliX
/dev/dri/cardX
DRM Core
DRM DSI Core (drm_mipi_dsi.c)
drm_dev_register
mipi_dsi_host_register
drm_p
Allwinner DE driver sun4i_drv.c
Rockchip DE driver rockchip_drm_drv.c
Allwinner DW HDMI sun8i_dw_hdmi.c
GPU Driver
Allwinner MIPI-DSI sun6i_mipi_dsi.c
Allwinner Display Controller
Rockchip Display Controller
Designware HDMI Controller
GPU HW
Allwinner MIPI-DSI Controller

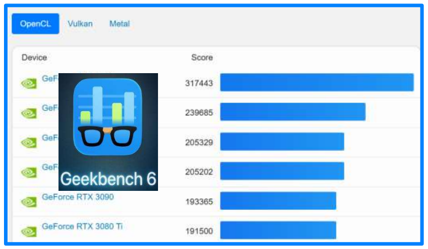
- Functional, performance and power validation of driver + IP not enough
- Application software on top of other SW layers and IP determine overall performance
- **Required**: Complete enough SW stack
- **Required**: Complete enough HW model – typically hybrid
- **Required**: Billions of cycles of software execution

Benchmarks need to Shift Left to have Impact

Virtual Platforms and Hybrid-HAV have successfully been used to Bring-Up SW and OS so far







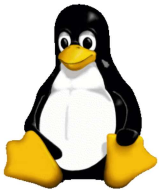
Device	Score
GeF	317443
GeF	239685
GeF	205329
GeF	205202
GeForce RTX 3090	193365
GeForce RTX 3080 Ti	191500

Too late to improve system

Pure Emulation is not Fast Enough

Example: Even a Linux boot takes more than 2 hours

Linux Boot Phase	Boot Time on Real Device (seconds) *	No. of Cycles @ 3 GHz	Wall Clock Time @ 3 MHz in sec
BIOS/UEFI Initialization	1	3 B	1,000
Bootloader	2	6 B	2,000
Kernel Loading	2	6 B	2,000
Kernel Initialization	3	9 B	3,000
Init/Systemd	2	6 B	2,000
Total	**10**	**30 B**	**2 hrs 46 min**



* How did we save time finding these numbers quickly?
Asked Copilot to produce a “close approximation of Redhat Linux 8.0 boot on ARM A-75 core at 3 GHz”

What is the Complexity of a Typical SW Stack

Example: Android Boot

Name	Wall Clock	Instruction Counter	Instruction Rate	Design Clock Cycles
U-Boot	00:00:48	7.9 M	0	771,053,384
Linux	00:00:56	370 M	47.8 M	785,993,541
Android Start	00:01:14	427 M	3.2 M	821,513,112
Android Complete	00:21:32	4.65 B	3.46 M	3,258,049,476
Home Screen	00:30:25	5.3 B	1.25 M	4,323,935,193

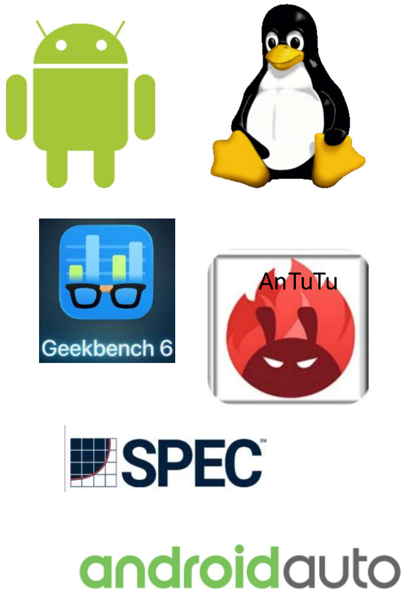
2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Application-level Hybrid Emulation
- Where to find the SW

accellera
SYSTEMS INITIATIVE

Open Source is Your Friend for SW Stacks

- Common use of open-source for OS stacks
  - Products use open-source operating systems and software stacks
  - Early access to software stacks → earlier system validation
- Freely available application Software
  - Apps and benchmarks are established in many markets
  - Easily added to pre-silicon environments
- Multitude of customer product configurations
  - Many configurations are defined by Software
  - Pre-silicon testing needs to include ability to change



How to use Open Source for OSes

- Download OS stacks like Linux and Android
  - https://kernel.org/
  - https://source.android.com/
- Configure OS'es to match your system maturity
  - https://www.mesa3d.org/ Open-source graphics implementation
- Configure the OS for an end-user ready system
  - Be capable to run workloads

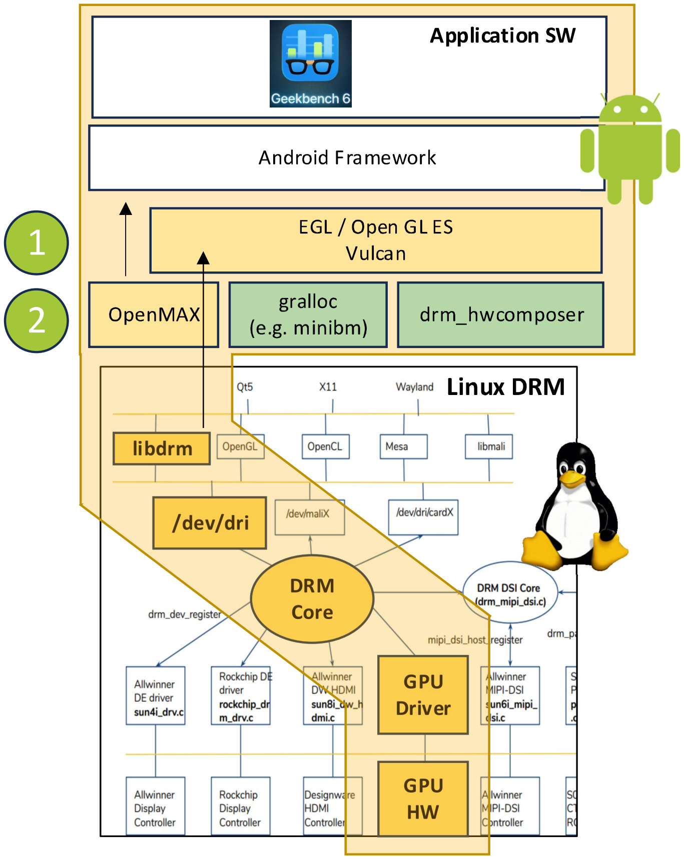
How to use Open Source for Application Benchmarks

- Standard end-user apps are freely available.
  - https://www.spec.org/cpu2017/
  - https://www.geekbench.com/
- Apps in source format: SPECCPU
- Apps in binary format: Geekbench
- Some apps require OS-configured shims
  - Compensate for missing functionality
  - https://docs.mesa3d.org/drivers/panfrost/drm-shim.html

Let's Talk About Android DRM

DRM allows multiple configurations of graphics

- Start with standard graphics and move to custom graphics, when available.
- Location for these properties
  - 1 /vendor/lib64/hw: gralloc.minigbm.so, hwcomposer.drm_minigbm.so
  - 2 /vendor/lib64/hw/egl: libGLES_mesa.so, libGLES_<vendor_name>.so
    - Property: ro.hw.egl=mesa
- Properties like ro.hw.egl=<vendor_name> define what libraries to use



Source: DRM_KMS_for_Android_v1

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Application-level Hybrid Emulation
- Enabling Technology

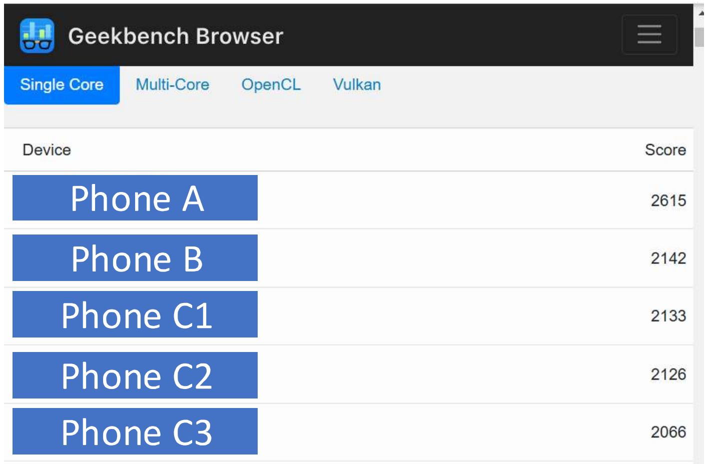
accellera
SYSTEMS INITIATIVE

Pre-Silicon Analysis Accuracy and Speed

- Pre-silicon workloads need accuracy
  - Pre-silicon metrics must be close to post-silicon

- Fast enough for single day TAT
  - HAV engines need to execute in the 5 MHz+ range
  - Virtual engines need to execute in 100 MHz+ range

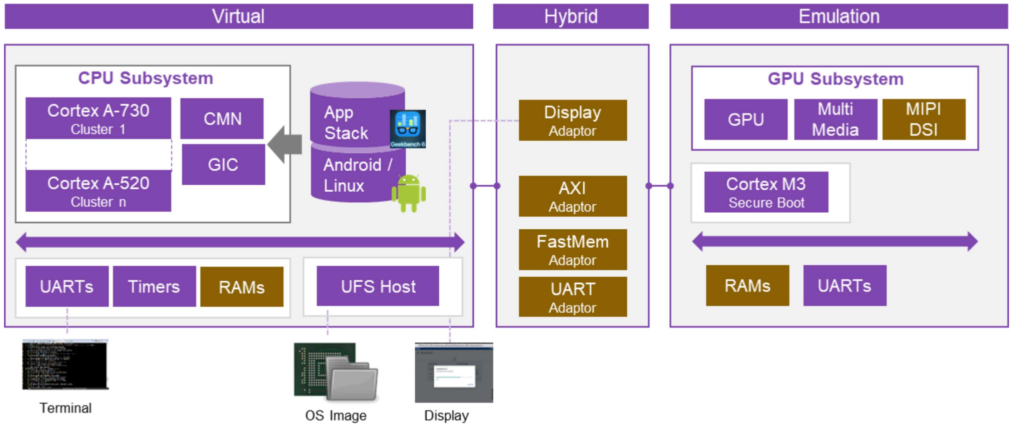
- Complete workloads with billions of cycles
  - Full workloads takes minutes of real-time



Device	Score
Phone A	2615
Phone B	2142
Phone C1	2133
Phone C2	2126
Phone C3	2066

https://browser.geekbench.com/android-benchmarks

Advanced Hybrid Run Benchmarks in 2 Days


Virtual

Hybrid

Emulation

CPU Subsystem

Cortex A-730 Cluster 1

Cortex A-520 Cluster n

CMN

GIC

App Stack

Android / Linux

UARTs

Timers

RAMs

UFS Host

Display Adaptor

AXI Adaptor

FastMem Adaptor

UART Adaptor

GPU Subsystem

GPU

Multi Media

MIPI DSI

Cortex M3 Secure Boot

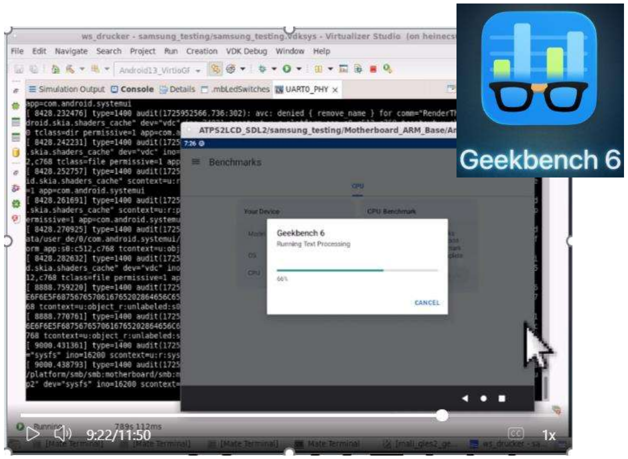
RAMs

UARTs

Terminal

OS Image

Display


ws_drucker - samsung_testing/samsung_testing.vdksys - Virtualizer Studio (on heinecs
File Edit Navigate Search Project Run Creation VDK Debug Window Help
Android13_VirtioGf
Simulation Output
Console
Details
.mbLedSwitches
UART0_PHY
ATPS2LCD_SDL2/samsung_testing/Motherboard_ARM_Base/Ar
7:26
Benchmarks
CPU
Your Device
CPU Benchmark
CANCEL
Running
789s 112ms
9:22/11:50
CC
1x
Geekbench 6

```
app=com.android.systemui
[ 8428.232476] type=1400 audit(1725952566.736:302): avc: denied { remove_name } for comm="RenderTh
droid.skia.shaders_cache" dev="vdc"
0 tclass=dir permissive=1 app=com.a
[ 8428.242231] type=1400 audit(1725
.skia.shaders_cache" dev="vdc" ino=
2,c768 tclass=file permissive=1 app
[ 8428.252757] type=1400 audit(1725
id.skia.shaders_cache" scontext=u:r
=1 app=com.android.systemui
[ 8428.261691] type=1400 audit(1725
.skia.shaders_cache" scontext=u:r:p
ermissive=1 app=com.android.system
[ 8428.270925] type=1400 audit(1725
ata/user_de/0/com.android.systemui/
orm_app:s0:c512,c768 tcontext=u:obj
[ 8428.282632] type=1400 audit(1725
d.skia.shaders_cache" dev="vdc" ino
12,c768 tclass=file permissive=1 ap
[ 8888.759220] type=1400 audit(1725
E6F6E5F6875676570616765202864656C65
68 tcontext=u:object_r:unlabeled:s0
[ 8888.770761] type=1400 audit(1725
6E6F6E5F6875676570616765202864656C6
768 tcontext=u:object_r:unlabeled:s
[ 9000.431361] type=1400 audit(1725
="sysfs" ino=16200 scontext=u:r:sys
[ 9000.438793] type=1400 audit(1725
/platform/smb/smb:motherboard/smb:m
p2" dev="sysfs" ino=16200 scontext=
```

Geekbench 6

Running Text Processing

66%

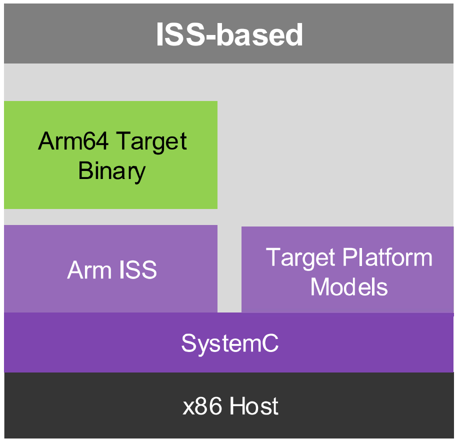
Virtual platforms boot Android **<10sec**, Geekbench in **11 minutes**

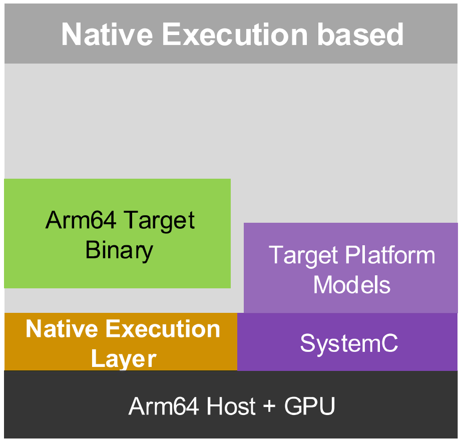
Hybrid platforms boot Android **<5min**, Geekbench in **2 days**

Emulation platforms boot Android **<40h**, Geekbench in **7 days**

Native Execution boost Virtual Prototypes Speed 100x+

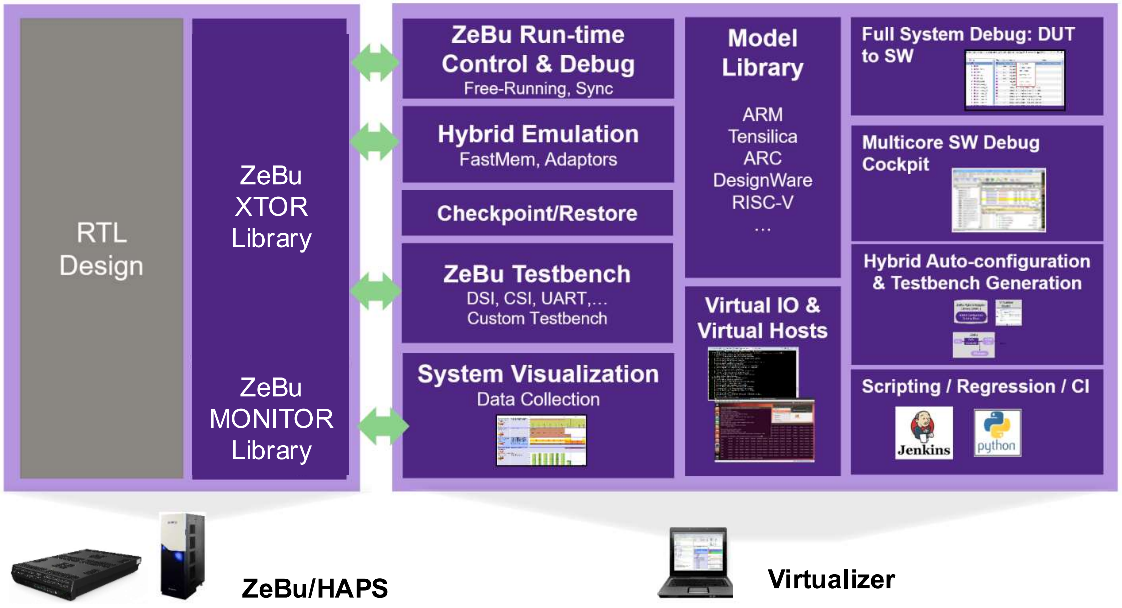
- JIT-based solutions (ArmFM, ImperasFPM or QEMU) on x86 take **10+ min** for Android boot
- Native execution on an Arm Server reduces boot time to **10s range**
- Key Requirements for Native Execution Layer
  - Ability to support CPU Parity
  - Tight connection with SystemC simulator
  - Tight connection with Hybrid Adaptor Layers





User Interaction for Hybrid from a Single Cockpit

- **Complete technology stack**, integration between ZeBu, HAPS-100 and Virtualizer
- **Best Productivity** for hybrid platform authoring
- **Best Performance** for runtime with seamless flow between ISS and native execution on Arm hosts.
- **Largest set** of pre-integrated **models** (Arm FastModels, Tensilica, RISC-V, CEVA & other 3rd party models)
- Integrated **System Level Debug** with major Software and Hardware debuggers



RTL Design

ZeBu XTOR Library

ZeBu MONITOR Library

ZeBu Run-time Control & Debug
Free-Running, Sync

Hybrid Emulation
FastMem, Adaptors

Checkpoint/Restore

ZeBu Testbench
DSI, CSI, UART,…
Custom Testbench

System Visualization
Data Collection

Model Library
ARM
Tensilica
ARC
DesignWare
RISC-V
…

Virtual IO & Virtual Hosts

Full System Debug: DUT to SW

Multicore SW Debug Cockpit

Hybrid Auto-configuration & Testbench Generation

Scripting / Regression / CI

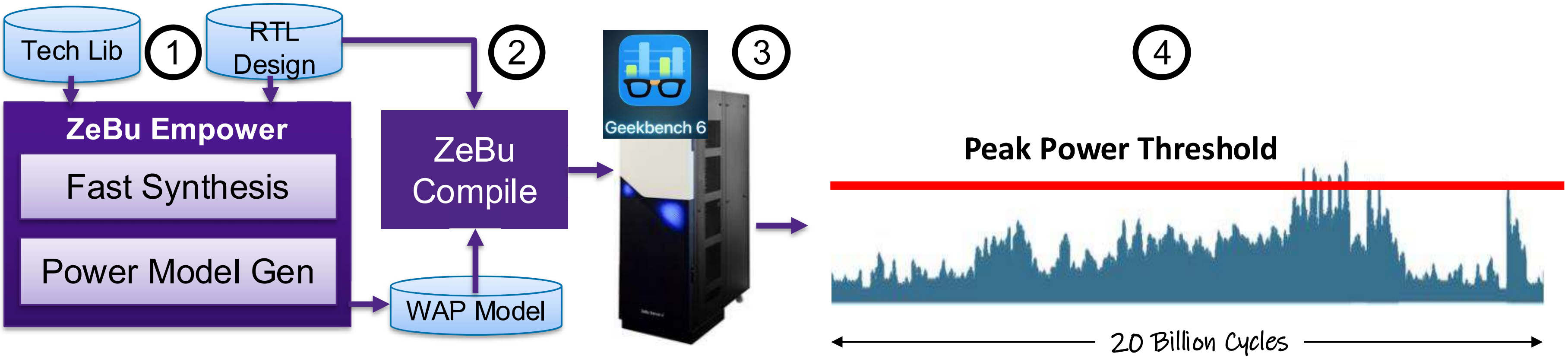
Jenkins

python

ZeBu/HAPS

Virtualizer

Fastest Power Analysis Technology



1. Design analysis to create Weighted Activity Model
2. Weighted Activity compiled into emulation model
3. Weighted Activity Profile (WAP) generated during emulation runtime
4. Analyze results in Verdi / GUI

Bet on the Highest Emulation Performance

- Highest performance for up to 5.8 BG designs
- 1.4 BG capacity per rack
- Proven HAV use cases
- Emulation and Prototyping configurability

ZeBu EP2

Insights from Different Engines used for Hybrid

	Insight
Virtual Prototype	Function traces across end-user apps to identify SW functions with adverse impacts on system performance
Emulator	Capture power across billions of cycles to find abnormalities: for example, high-power with low performance

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

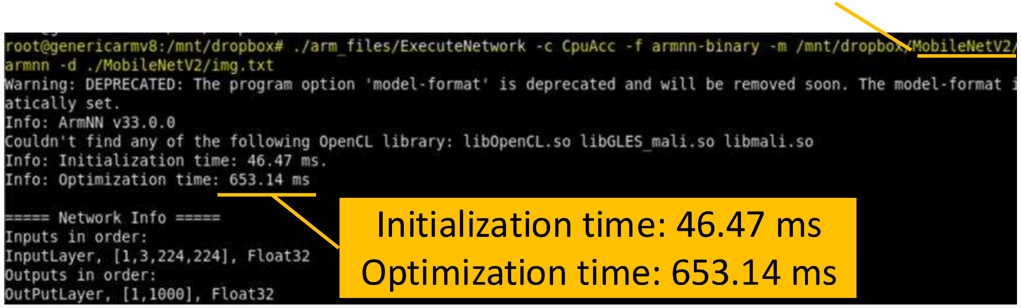
Application-level Hybrid Emulation - Examples

accellera
SYSTEMS INITIATIVE

AI Performance Insights from MobilenetV2

- MobilenetV2 classifies input images into one of 1,000 classifications
- MobilenetV2 need to achieve power and perf specs:
  - Initialization time
  - Optimization time
  - Inference time
- Image # 945 = Bell Peppers ☑

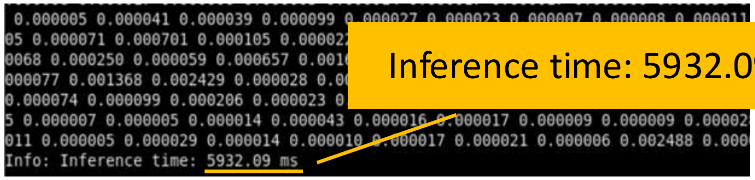
MobileNetV2



```
root@genericarmv8:/mnt/dropbox# ./arm_files/ExecuteNetwork -c CpuAcc -f armnn-binary -m /mnt/dropbox/MobileNetV2/
armnn -d ./MobileNetV2/img.txt
Warning: DEPRECATED: The program option 'model-format' is deprecated and will be removed soon. The model-format i
atically set.
Info: ArmNN v33.0.0
Couldn't find any of the following OpenCL library: libOpenCL.so libGLES_mali.so libmali.so
Info: Initialization time: 46.47 ms.
Info: Optimization time: 653.14 ms

===== Network Info =====
Inputs in order:
InputLayer, [1,3,224,224], Float32
Outputs in order:
OutPutLayer, [1,1000], Float32
```

Initialization time: 46.47 ms
Optimization time: 653.14 ms

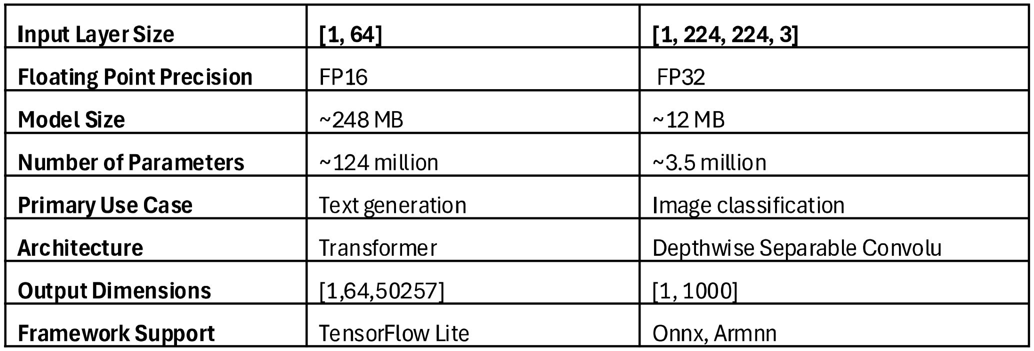


```
 0.000005 0.000041 0.000039 0.000099 0.000027 0.000023 0.000007 0.000008 0.000011
05 0.000071 0.000701 0.000105 0.00002
0068 0.000250 0.000059 0.000657 0.001
000077 0.001368 0.002429 0.000028 0.0
0.000074 0.000099 0.000206 0.000023 0
5 0.000007 0.000005 0.000014 0.000043 0.000016 0.000017 0.000009 0.000009 0.00002
011 0.000005 0.000029 0.000014 0.000010 0.000017 0.000021 0.000006 0.002488 0.000
Info: Inference time: 5932.09 ms
```

Inference time: 5932.09 ms

[List of Imagines: Deeplearning User Guide]

Performance Insights From GPT2 LLM



	GPT2 LLM ☒	MobileNetV2 ☑
Input Layer Size	**[1, 64]**	**[1, 224, 224, 3]**
Floating Point Precision	FP16	FP32
Model Size	~248 MB	~12 MB
Number of Parameters	~124 million	~3.5 million
Primary Use Case	Text generation	Image classification
Architecture	Transformer	Depthwise Separable Convolu
Output Dimensions	[1,64,50257]	[1, 1000]
Framework Support	TensorFlow Lite	Onnx, Armnn

Issue for larger GPT2 LLM

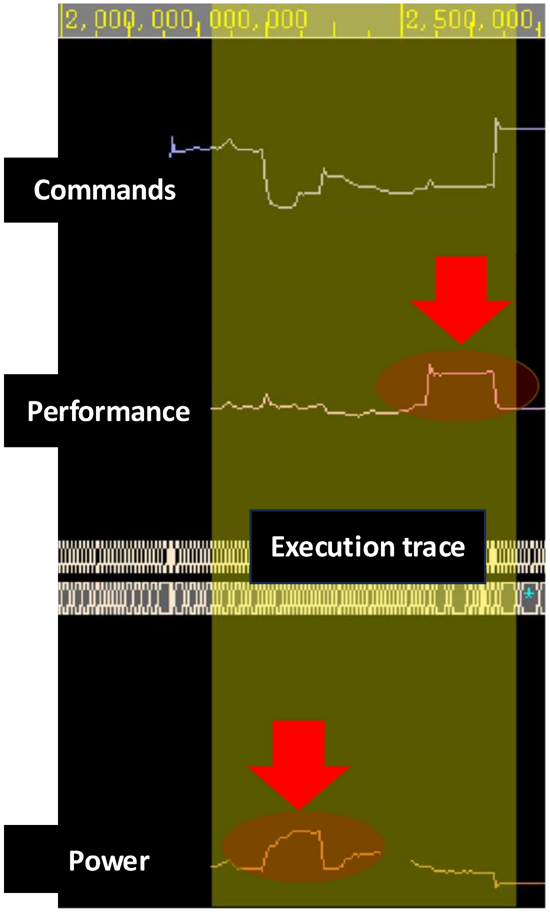
- AXI ID width was too small for larger data transfer needs
- The ID's for the larger models were compromised, creating errors

Executing GPU Benchmark Exposes Design Issues

- High-power and high-performance events are not always correlated

- Performance and Power must be extracted for full benchmark to get optimum system

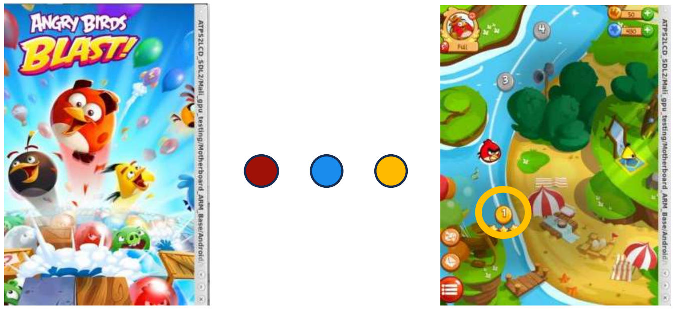
GFX Bench



Need to Quickly Get to Interesting Area

- Start of the application isn't "interesting" since it goes through initialization

- It takes seconds of real-time to real application is executing (Billions of cycles)

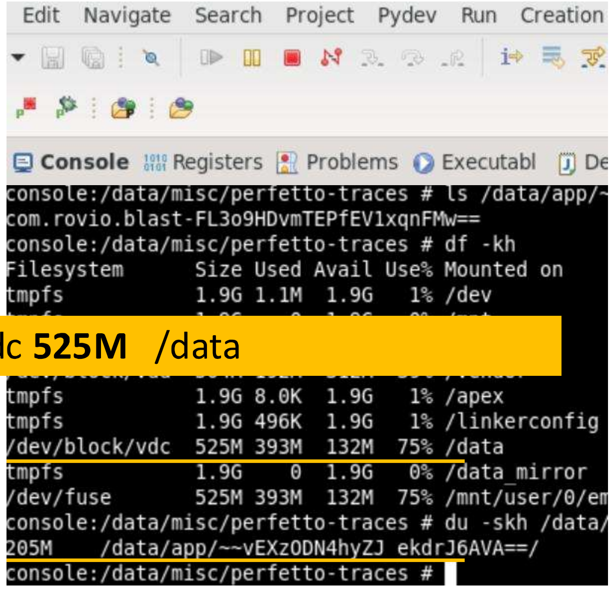


Start

Area of interest

Need End-User Ready System

- You need to acquire, install, run Apps
- Apps are stored into Android's /data fs
  - This example shows /data has 525M of space
- To install the system needs enough space
  - Anry Birds needs 205 M (**installed in /data**)
  - Geekbench needs 700 M (**too big for /data**)


Edit Navigate Search Project Pydev Run Creation
Console Registers Problems Executabl De

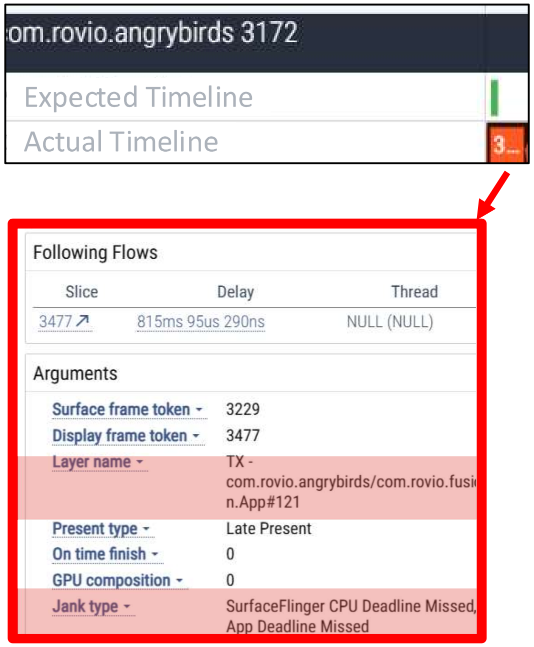
```
console:/data/misc/perfetto-traces # ls /data/app/~
com.rovio.blast-FL3o9HDvmTEPfEV1xqnFMw==
console:/data/misc/perfetto-traces # df -kh
Filesystem       Size Used Avail Use% Mounted on
tmpfs            1.9G 1.1M  1.9G   1% /dev
tmpfs            1.9G 8.0K  1.9G   1% /apex
tmpfs            1.9G 496K  1.9G   1% /linkerconfig
/dev/block/vdc   525M 393M  132M  75% /data
tmpfs            1.9G    0  1.9G   0% /data_mirror
/dev/fuse        525M 393M  132M  75% /mnt/user/0/em
console:/data/misc/perfetto-traces # du -skh /data/
205M    /data/app/~~vEXzODN4hyZJ ekdrJ6AVA==/
console:/data/misc/perfetto-traces #
```

/dev/block/vdc **525M** /data

205M /data/app/~~ .. Angry

Using Android Perfetto for System Insights

- How well does the interconnection and memory work?
- Actual Timeline to create frames is longer than Expected Timeline
- Software Layer: *rovio.angrybirds*
- Graphics exhibits shows ‘jank’ problem
- Perfetto Jank Type: *CPU Deadline Missed* ☒



om.rovio.angrybirds 3172

Expected Timeline

Actual Timeline

Following Flows

Slice	Delay	Thread
3477 ↗	815ms 95us 290ns	NULL (NULL)

Arguments

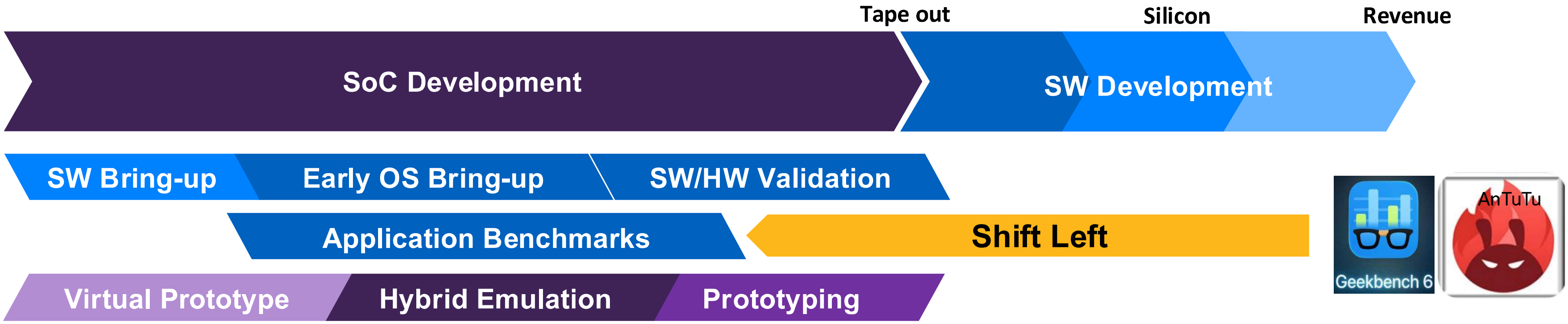
Surface frame token	3229
Display frame token	3477
Layer name	TX - com.rovio.angrybirds/com.rovio.fusi n.App#121
Present type	Late Present
On time finish	0
GPU composition	0
Jank type	SurfaceFlinger CPU Deadline Missed, App Deadline Missed

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Application-level Hybrid Emulation - Outlook

accellera
SYSTEMS INITIATIVE

Lookout DV teams: This is coming !!!



Tape out

Silicon

Revenue

SoC Development

SW Development

SW Bring-up

Early OS Bring-up

SW/HW Validation

Application Benchmarks

Shift Left

Virtual Prototype

Hybrid Emulation

Prototyping

Geekbench 6

AnTuTu

- Software stacks beyond OS becoming more important
- Pre-silicon application benchmarks becoming next sign-off
- Faster execution platform performance is the key enabler
- Let’s learn and develop new skills and methodologies together

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

Thank You for Your Interest

accellera
SYSTEMS INITIATIVE