Unveiling Advance Hybrid Emulation Methodology for Accelerated Android Home Screen Bring-up and System Level Verification

Rinkesh Yadav(rinkesh.y@samsung.com)
Sarang Kalbande(sarang.m@samsung.com)
Hyundon Kim(hyundon.kim@samsung.com)
Garima Srivastava(s.garima@samsung,com)

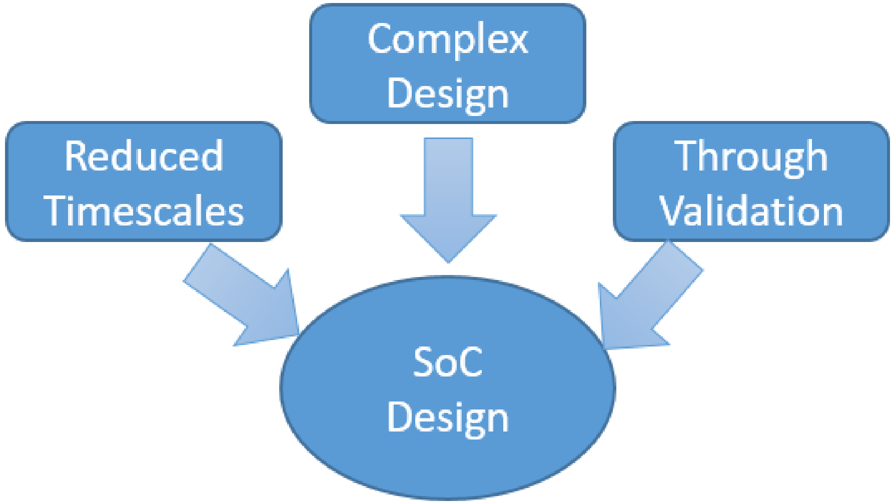
Samsung Semiconductor India Research

Agenda

- High performance Emulation methodology: Need & Impact
- Challenge to bring-up software on Pure Emulation
- Hybrid Emulation : Introduction
- Easy Portability from Pure to Hybrid Emulation
- Results and Benefits : Emulation v/s Hybrid Emulation
- Future Scope : CPU Benchmark
- Acknowledgement & References

High performance Emulation methodology: Need & Impact

- ❑ Industry need to Launch Product constantly faster in market with **higher performance** and **less power greedy.**
- ❑ Increased **design complexity** and **shrunk time to market**, not an easy to launch product on time to competitive others.

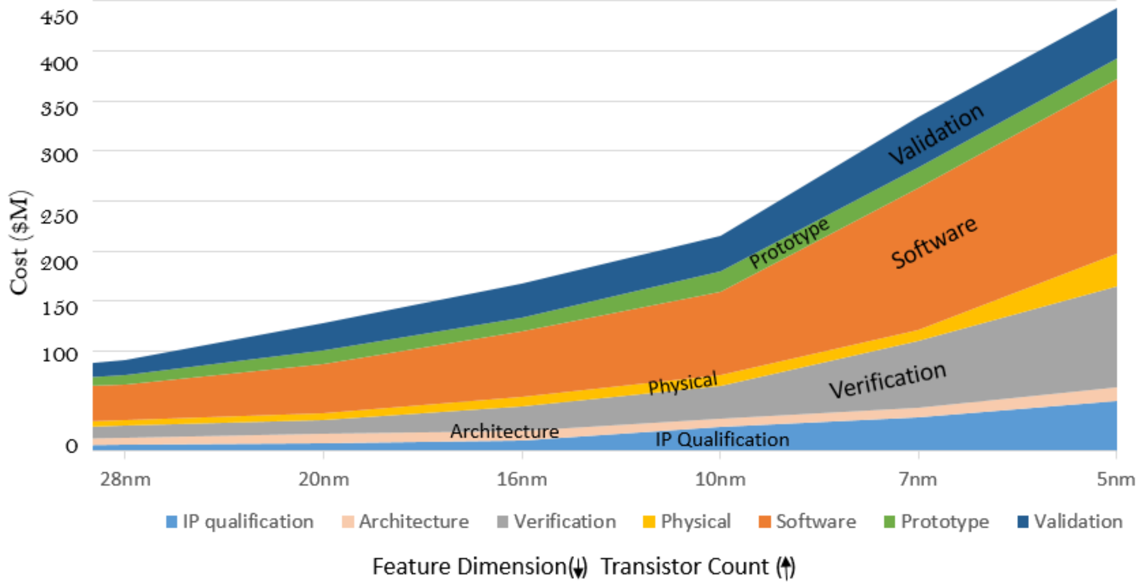


Complex Design

Reduced Timescales

Through Validation

SoC Design



- ❑ SoC Software development and validation done at post-silicon which could effects time to market and cost.

High performance Emulation methodology: Need & Impact(2)

- Is Simulation viable for SoC Software development ?

  Though Good design debug visibility but runtime is a bottleneck

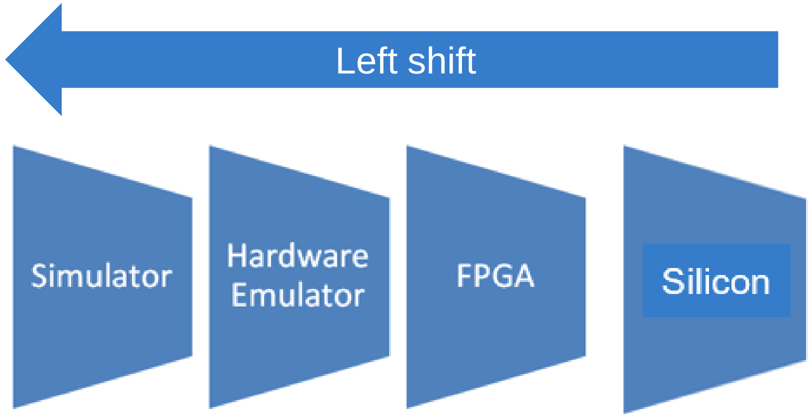
- Is Traditional Emulation a solution for SoC Software Development?

  Improved Runtime but still takes time to develop SoC software and system tests

- Is FPGA and Silicon a solution ?

  It has Less design debug visibility ,high cost ,Respin cost.

- Left Shift and High Run time performance.



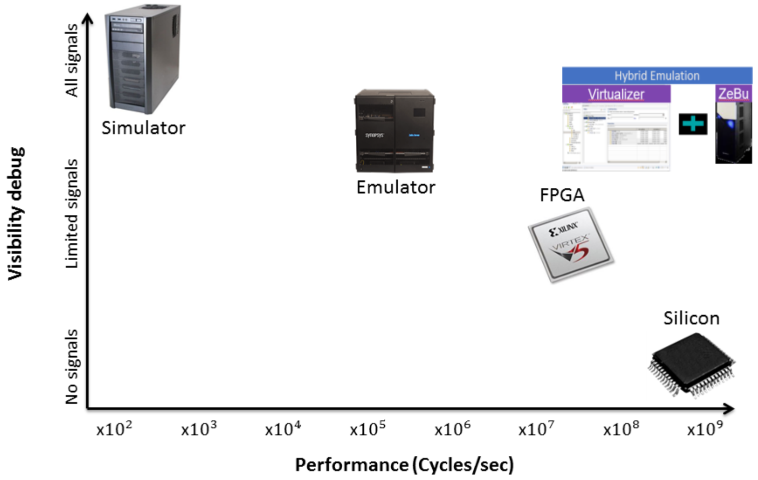
Left shift

Simulator

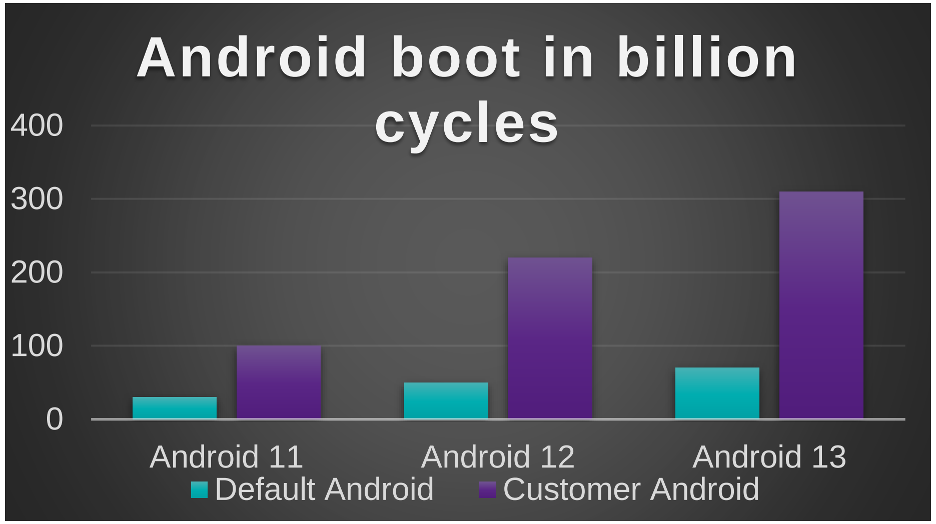
Hardware Emulator

FPGA

Silicon



Challenge to bring-up software on Pure Emulation



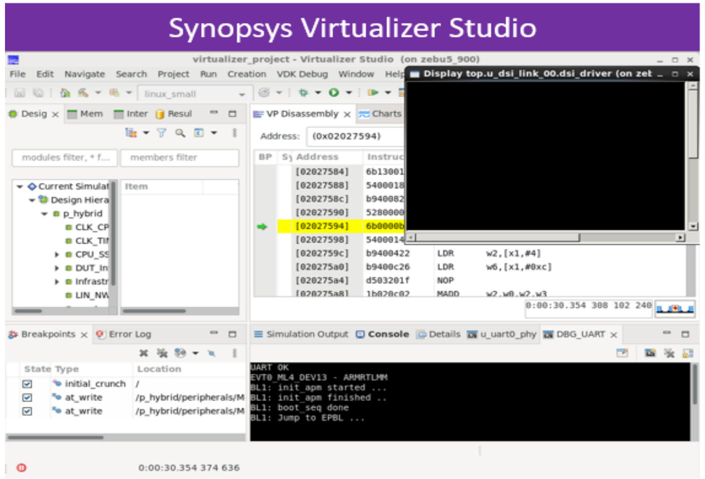
android 13 boot time(cpu @3Mhz, 4cycles/ instruction):

(310Bx4/3 MHz)/3600s = 114 hours (~ 4 days)

- ❑ Emulator driver clock frequency is not sufficient to bring-up Linux kernel, Android OS boot and to develop system level software.
- ❑Takes 114 hours to bring-up Android OS where SoC is running on 3Mhz emulator driver clock frequency.
- ❑ To overcome less emulator speed and to accelerate software development, Hybrid emulation is introduced.

Hybrid Emulation: Introduction

- Hybrid emulation combines Virtualizer and Hardware emulator.
- one part of the SoC design is run at the emulator and the other part is run at virtual platform.
- Virtual prototypes are high performance, System C models of a particular block, a system model or an entire SoC as per requirements.
- The task of virtual platform is to have enough accuracy to support the level of software being run on it. This is achieved by modelling the behaviour and inter-block communication at transaction level(TLM), which makes these faster than equivalent cycle-accurate RTL.


Synopsys Virtualizer Studio

virtualizer_project - Virtualizer Studio (on zebu5_900)
File Edit Navigate Search Project Run Creation VDK Debug Window Help
Display top.u_dsi_link_00.dsi_driver (on zet
VP Disassembly
Charts
Breakpoints
Error Log
Simulation Output
Console
Details
u_uart0_phy
DBG_UART

Address: (0x02027594)

```
[02027584]  6b13001
[02027588]  5400018
[0202758c]  b940082
[02027590]  5280000
[02027594]  6b0000b
[02027598]  5400014
[0202759c]  b9400422   LDR    w2,[x1,#4]
[020275a0]  b9400c26   LDR    w6,[x1,#0xc]
[020275a4]  d503201f   NOP
[020275a8]  1b020c02   MADD   w2,w0,w2,w3
```

0:00:30.354 308 102 240

State	Type	Location
☑	initial_crunch	/
☑	at_write	/p_hybrid/peripherals/M
☑	at_write	/p_hybrid/peripherals/M

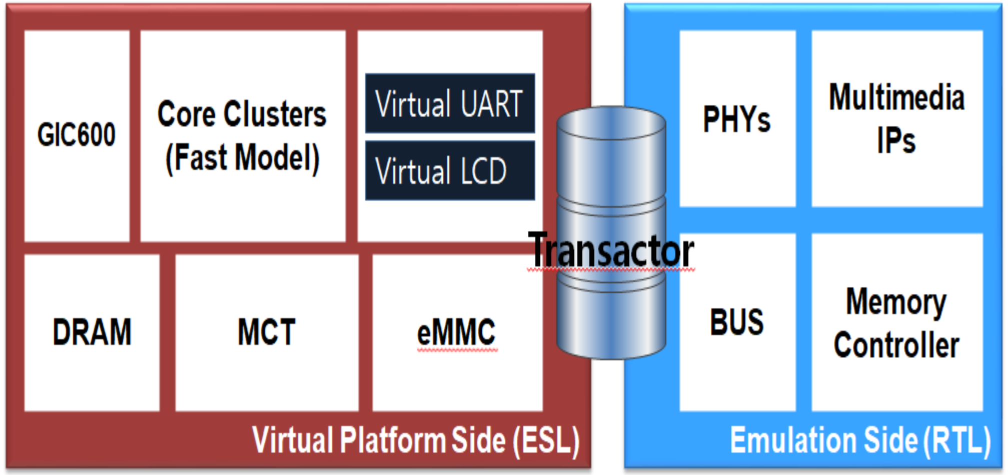
```
UART OK
EVT0_ML4_DEV13 - ARMRTLMM
BL1: init_apm started ...
BL1: init_apm finished ..
BL1: boot_seq done
BL1: Jump to EPBL ...
```

0:00:30.354 374 636

+

ZeBu

=



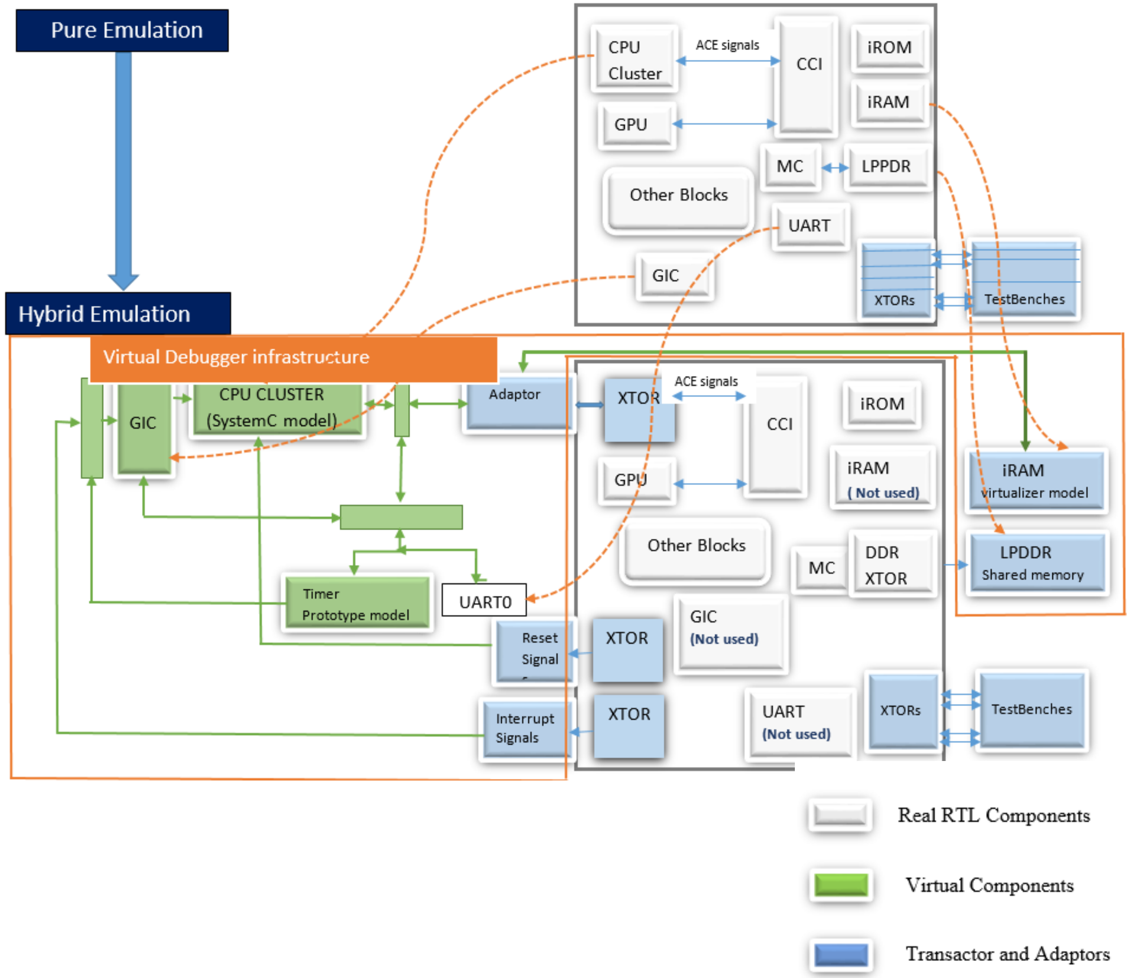
Easy Portability from Pure to Hybrid Emulation

❑ @ Linux/Android boot, many AXI/CHIE bus transactions at **cpu - dram** path.

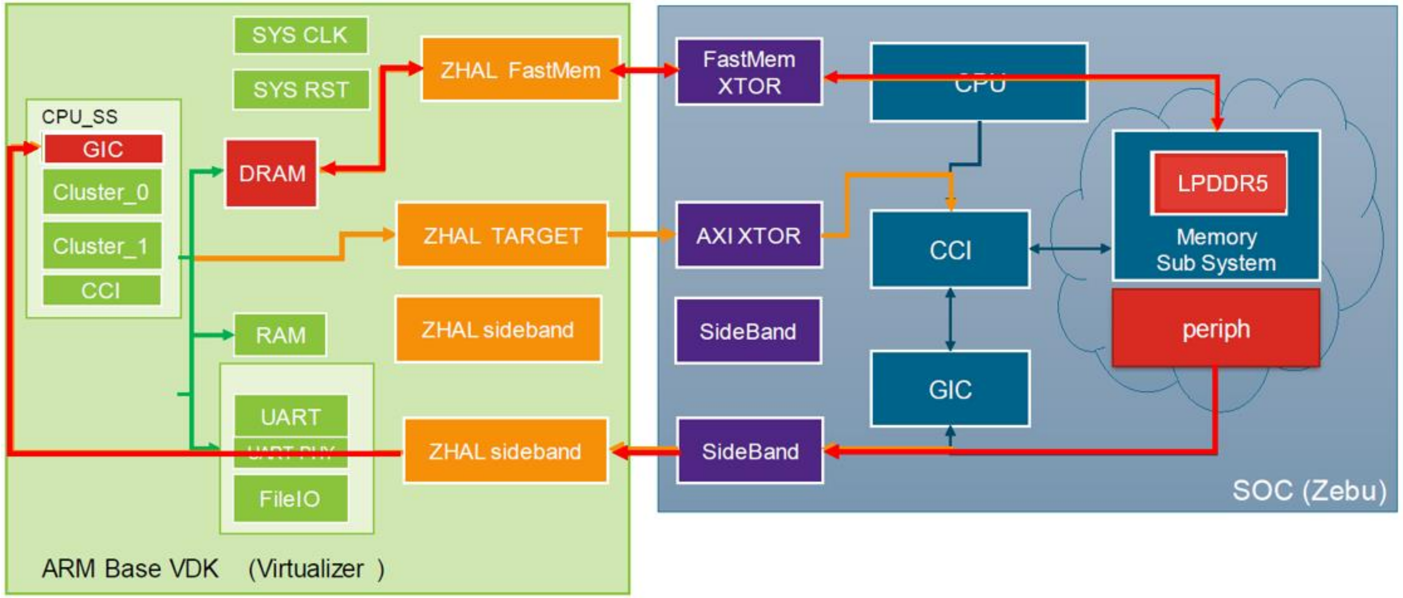
❑ **High latency** cpu dram transactions are bottle-neck for pure emulator performance to **SW development.**

❑ To overcome this, **CPU and related components** moved to **virtual side** for **OS boot** faster and accelerative.

❑ Acceleration achieved because of virtual model of the CPU which runs **Instruction Set Simulation** (ISS) to perform much faster compared to CPU's **gate level behavior** in emulator.



Easy portability from Pure to Hybrid Emulation(2)



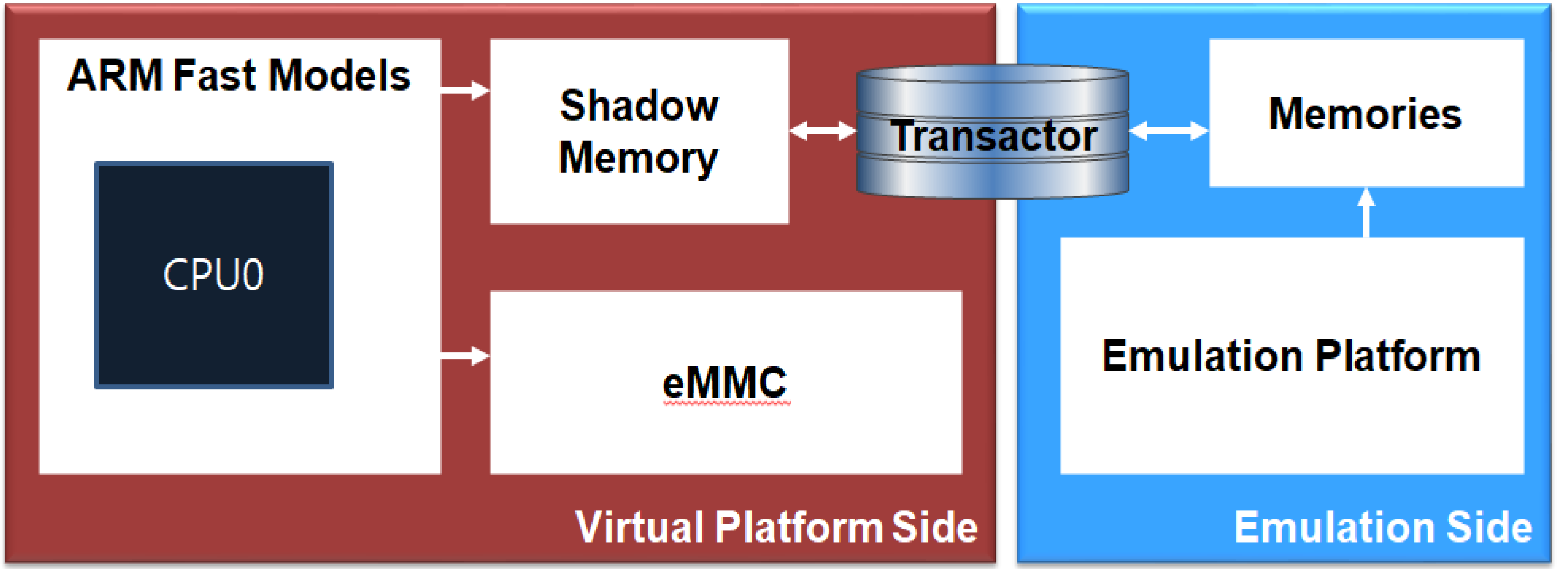
To build the hybrid Virtual Platform:

- ❑ **Block/IP** mapped on **ZeBu** have to be **remove** or **disable.**
- ❑ Connection has to be made on the interface with ZeBu:
  - **AMBA Busses**
  - **Sideband Signals**
  - **ZeBu shared memory**
- ❑ Hybrid Adaptor provides a library of wrappers (**ZHAL**) for these components.
- ❑ It can be used inside virtualizer as any TLM model.

Easy Portability from Pure to Hybrid Emulation(3)

- ❑Hybrid emulation has **Fastmem model** of DRAM instead of **RTL memory model**.

- ❑Hybrid fast memory is **shared memory** between **virtualizer** and **emulator** which is properly **synced** between them.

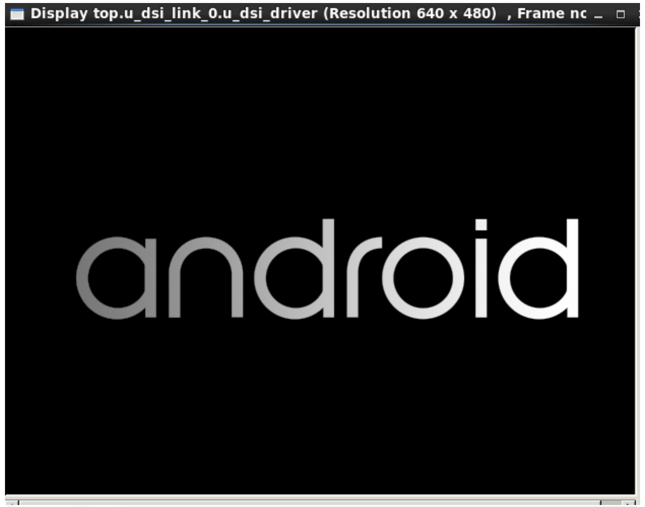


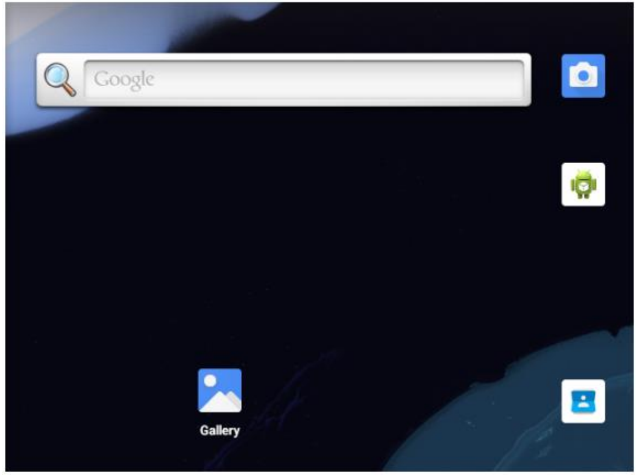
Results and Benefits: Emulation V/S Hybrid Emulation

@Exynos SoC

	Simulation	Pure Emulation	Hybrid Emulation
Environment Intitialization	4 Min	5 Min	5 Min
Kernel Boot-up(prompt)	125,865 Min	400 Min	2 Min
Android Logo	231,000 Min	1200 Min	12 Min
Android Home Screen	510,517 Min	3200 Min	53 Min
Total Consumed Time	867,384 Min	~4805 Min	~72 Min
Clock Frequency	866 Hz	3 Mhz	x20 → ~59.9 Mhz

Unrealistic


Display top.u_dsi_link_0.u_dsi_driver (Resolution 640 x 480) , Frame nc
android



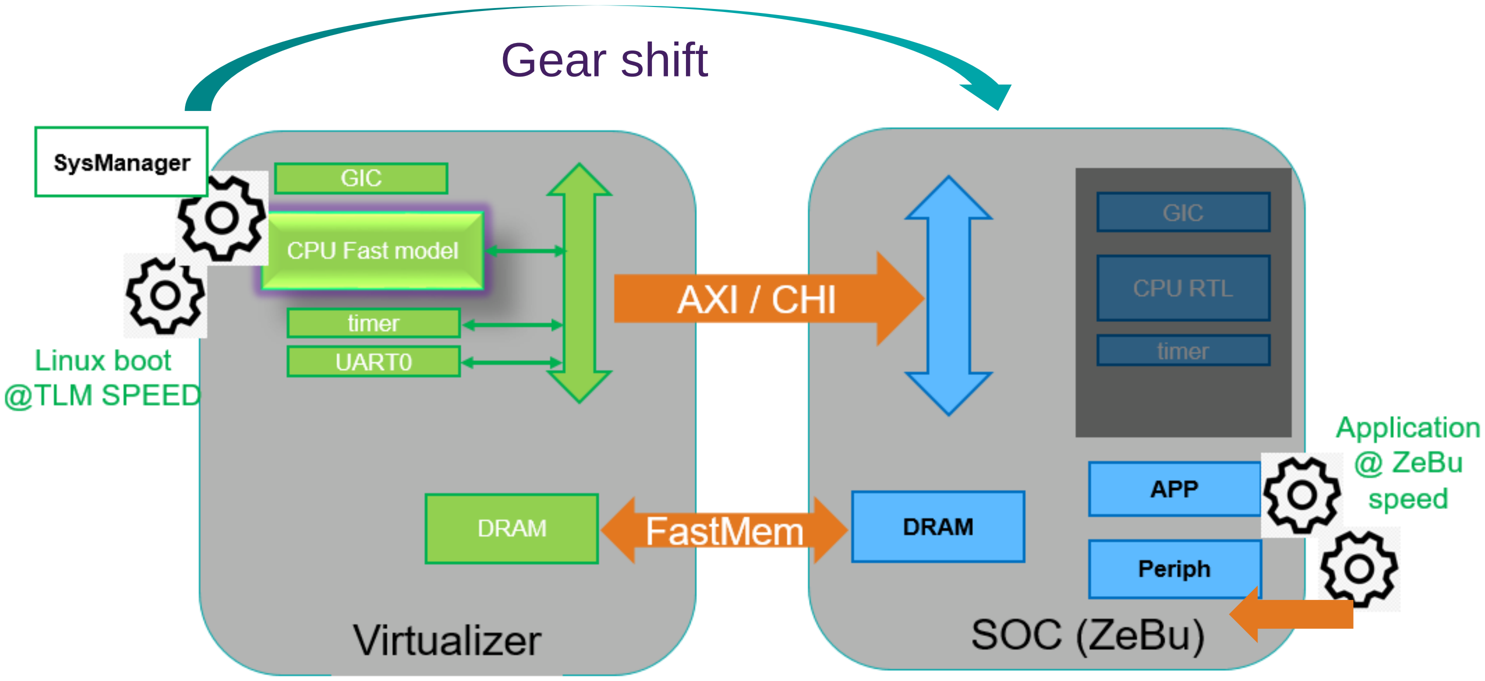
- ❑ **Linux** brought-up in **2 Minutes**.
- ❑ **Android Logo** brought-up in **12 minutes**.
- ❑ **Android Home Screen** brought-up in **72 Minutes**.

Results and Benefits: Emulation V/S Hybrid Emulation(2)

Benefits:

- ✓ Early Bring-up **Linux kernel** and **Android home screen**.
- ✓ Hybrid Emulation can be use **for firmware development** and **Device driver bring-up** at pre-silicon stage like GPU, DPU, AUDIO etc.
- ✓ **Higher debug capability** with various debugger methods support like:
  - **TLM transaction** debug through **Chart View**
  - **VP disassembly** feature and **t32** debugger to debug software
  - Support different **log levels** to debug **Transactor and Adaptor**
  - Support **QWIC/FWC** waveform dump capture to debug **hardware**
- ✓ **Validate Software** use cases for all **IPs** modelled in **RTL**.
- ✓ Identify **Early SW bugs**.

Future Scope: CPU Benchmark



- Once **Linux/Android** boot is done, **CPU STATE** will left shift from virtual to **RTL CPU** running in **ZeBu**.

- As Now **Linux/Android** is already **booted**, **CPU Benchmark** can be run on **Pure Emulation**.

Acknowledgement

- Vievkananad Vivek (v.vivek@samsung.com)
- Hwan-sung PARK (hs43.park@samsung.com)
- Synopsys Support Team

References

- How FPGA boards help to validate ARM processors (https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/how-fpga-boards-help-to-validate-arm-processors) [Slide 1-2]
- Synopsys Hybrid Emulation Documents and Quick Start User Guide.

SAMSUNG snug

THANK YOU

Our
Technology,
Your
Innovation™

Q&A