+ + + + + + + + + + + + +

Arm Development Studio

drm d

+ + + + + + + + + + + + + + +

+ + + + + + + + + + + + + +

* * * * * * * * * * * * * * * *

Learning Outcomes

At the end of this module, you will be able to:

Arm Development Studio overview

* * * * * * * * * * * * * *

Arm Development Studio featuring Keil MDK

Complete suite of tools for Arm Cortex-A/R/M based devices



Earliest Arm core support





Accelerates time to market

Everything you need to develop Arm based projects from tiny sensors to server grade multi-core SoCs

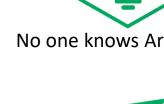
- Includes Arm Compiler 6, Debugger, Streamline performance analyzer, Mali Graphics Debugger and Fixed Virtual Platforms
- Includes Keil MDK for power optimized microcontroller software development
- Supports all SoC configurations: from single core to complex multi-cluster and multi-core

Investment protection

- Early and dependable support for new Arm hardware IP
- Used in Arm for architecture validation
- Developed over many years of continuous improvement with billions of shipped products built
- Technical support from the experts in Arm technology

Innovate earlier and develop faster for competitive edge

- Develop software pre-silicon
- Supports full development workflow from emulators to production hardware
- Software building blocks for fast application development including CMSIS, middleware and RTOS
- Targeted performance analysis for rapid software optimization



Arm Development Studio



Create & build Code

- Latest Arm processor support
- Optimized code generation and
 - embedded libraries



- Debug Code
- Inspect CPU state and execution
- Set breakpoints, run scripts, debug exceptions
- Cache/MMU/MPU visibility
- AMP, SMP, big.LITTLE, DynamIQ



Trace Capture Device Source Ranges Detail UXTH r1,r5 MOVS r0,#0x50 wr_reg ; 0x80011B8 0×08000E68 ${r4-r6, lr}$ PUSH

Visualize Code

- View code history up to breakpoint
- View code instrumentation output
- Make sense of OS level events and threads
- Track GPU calls and reconstruct frame buffers

Speed Up Code

- Tune code for optimal cache usage
- Thread/process level visualization
- GPU usage vs CPU usage
- Energy use over time



Fast & Cycle Models

• Virtual platforms for architectural exploration and early software development

Tools designed for the entire product lifecycle



DSTREAM™

wr_reg

wr_cmd <Unknown:

GLCD_Clear ADC Init

SystemCoreClockUpdate

Address 0x08000E64

0x08000F66

• Family of debug hardware for FPGA & board bring-up for complex SoCs



ULINKpro D

• Family of low-cost debug hardware for MCU & MPU



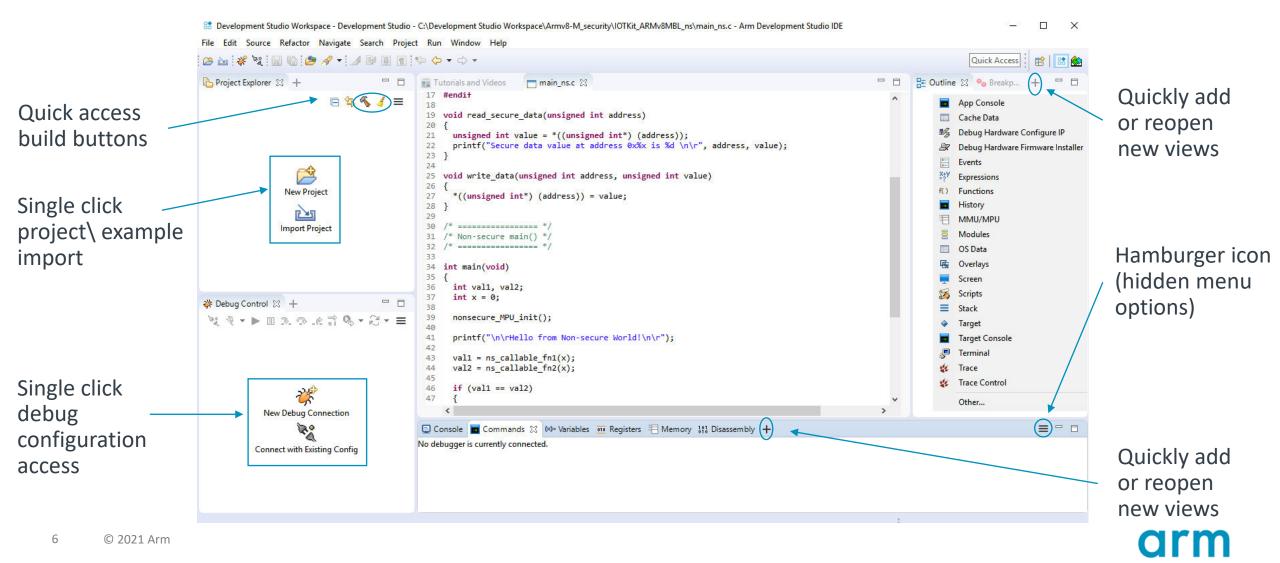
TCP/IP

• Application-level debug for BSP and reference design rollout



Arm Development Studio IDE

One perspective, simplified UI with easy and quick access



Keil MDK overview

arm KEIL

| Tools | MDK | | Arm | C/C++ | Compiler |
|-----------|---|------------------------------|----------------------|----------|------------------------------|
| MDK Tools | μVision IDE
with pack m | | With | safety o | qualification |
| | Device | CMSIS | MD | K-Mid | ldleware |
| Packs | Startup | CMSIS-Core | Network
IPv4/IPv6 | | Graphics |
| Software | Device HAL | CMSIS-DSP | USB
Host/Devic | ce | Mbed TLS
SSL/TLS security |
| Sof | CMSIS Drivers | CMSIS-RTOS | File Syster | m | IoT connectors |
| Probes | ULINK <i>pro</i>
for code coverage ana | ULINK
Ilysis for power op | | for ru | ULINK2
un/stop debugging |

drm Arm Compiler 6

* * * * * * * * * * * * * * * * *

+ + + + + + + + + + + + +

+ + + + + + + + + + + + + + + + +

+ + + + + + + + + + + + + + +

+ + + + + + + + + + + + +

Arm Compiler 6: Leading-edge C/C++ toolchain



Comprehensive support





- The only embedded compiler you'll ever need
 - Validated support for new Arm technologies significantly
 - Architecture validation
 - Developed and supported by Arm architecture experts
- Efficient and accurate code for bare-metal and RTOS
 - Finely tuned Arm proprietary embedded libraries and linker
 - Optimized for a wide range of applications, not just a single benchmark
 - Support for the most recent language standards
- Path to functional safety certification
 - TÜV SÜD qualified for software development to multiple safety standards
 - Comprehensive Qualification Kit provides essential supporting documentation
 - Long Term Maintenance from stable & isolated code branch

Basic armclang options

armclang -c --target=arm-arm-none-eabi -mcpu=Cortex-A9 -g -O2 foo.c

- --target= selects the execution state
 - arch64-arm-none-eabi for v8A AArch64
 - arm-arm-none-eabi for v8A AArch32, Cortex-M, and Cortex-R, and older CPUs
- Additionally require
 - -march and/or -mcpu to select specific architecture or core
 - -mcpu=list shows all available options (for specified target)
- −On to select optimization level (0≤n≤3)
 - Also -Os, -Oz, -Ofast, -Omax
 - -00 default (extremely poor optimization)
 - -flto to enable link-time optimization (implied with -Omax)
- -g to enable debug info

Basic armlink options

- --cpu= selects the CPU (or architecture)
 - Mainly used as a check when merging projects
- --scatter to specify scatter description file
 - Defines platform memory map
- --entry to specify entry point (symbol or address)
- --lto Link-Time Optimization (armclang objects built with -flto)

Arm Compiler 5 (armcc)

- Legacy compiler, no longer in development
 - No support for any Armv8 architectures
 - However 20+ years of history, many projects exist based on this compiler
- Recommend all projects start with (or migrate to) Arm Compiler 6 (armclang)
 - Thorough migration <u>documentation</u> available
 - For many users, C/C++ source code will simply be directly portable
 - Changes only to command options
 - armcc source containing embedded assembler must be re-written to GNU assembler (gas) format
 - Recommend rewriting with compiler <u>intrinsics</u>



Debug and trace

* * * * * * * * * * * * * * *

+ + + + + + + + + + + + +

Arm Debugger key points

Full support of Arm IP

- Arm Debugger is developed alongside Arm IP
- Support for all Arm cores, including leading-edge support for cores that are not yet public
- Comprehensive support for all Arm debug and trace IP

Flexible architecture

- Support for target complexity, challenging implementations, non-Arm IP
- Extension APIs customize debugger functionality, integrate with other tools

Designed for productivity

- CMSIS-Pack compatibility
- Integrated with Eclipse and Python – both widely used
- Debugger views and functionality optimized for fast, efficient, and clear operation
- Advanced bring-up tools, expert support world-wide
- Comprehensive scripting support with shallow learning curve

arm

Comprehensive support for Arm cores

- Arm Development Studio is developed alongside Arm cores
- Debug support for cores and architectural extensions is developed as soon as it is practical
- Development Studio is used as part of the normal core development and validation process
- Debugger testing is carried out against Fast Models and early FPGA implementations of all Arm cores
- Tools available for leading-edge partners working with future cores which are not yet public

| ARM
Cortex [®] -A75 | |
|--|--|
| ARM CoreSight™ | Multicore Debug and Trace |
| ARMv8.2-A
32b/64b CPU | Core 1
NEON [™]
SIMD engine
Floating
point unit |
| 64kB LI I-cache w/Parity | 64kB LI D-cache w/ECC |
| 256kB-512kB Pr | ivate L2 w/ECC |
| SCU Snoop Async
Filter Bridges | Shared L3 cache (512kB~4MB)
w/ECC |
| ACP Peripheral
Port | AMBA [®] 4 ACE or AMBA 5 CHI |

Advanced User Interface Design

Simplicity, visibility, and performance

| Register Set: My Reg | isters | | |
|----------------------|--------------------|------|--------|
| Name | Value | Size | Access |
| 🖻 🗁 Core | 5 of 47 registers | | |
| - 👄 R0 | 0x000000AA | 32 | R/W |
| – 🖌 R1 | 0x000001CC | 32 | R/W |
| – 🛛 R2 | 0x00000000 | 32 | R/W |
| - 💿 SP | 0x800C9578 | 32 | R/W |
| ⊢ ⊚ PC | 0x80000486 | 32 | R/W |
| 🖻 📂 CP15 | 3 of 134 registers | | |
| 🖻 🗁 MMU | 3 of 32 registers | | |
| 🖶 🦠 TLBTR | 0x00000400 | 32 | RO |
| - 🕤 TTBRO | 0x80500000 | 32 | R/W |
| 🗆 🕤 TTBR1 | 0x00000000 | 32 | R/W |

| 💊 Breakpoints 竝 Registers 🔲 | OS Data | Tasks X+V Expressions | ⊠ fO | Functions + | |
|-----------------------------|---------|-------------------------|------|--------------|--------|
| | | | | 0) | < 🗙 🕂 |
| Name | Value | Туре | Size | Location | Access |
| UART0->UARTFR & 0x10 | 16 | volatile const uint32_t | 32 | | RO |
| – UARTO->UARTDR | 0 | volatile uint32_t | 32 | S:0x1C090000 | R/W |
| 🕀 🥎 line | | char* | 32 | \$R5 | R/W |
| – ⊜ cnt | 0 | int | 32 | \$R6 | R/W |
| Enter new expression here | | | | | |

| ≡ Stack ∑ | ≍ `+ | | | (| ×)= (] 🕎 | ∲ ≡ - | ° 🗆 |
|---|---------------|----------------|-------|------|----------|--------|-----|
| Function P | rototype | Source/Line | | | | | |
| _svcM | essageGet() | rt_CMSIS.c:181 | 7 | | | | |
| osMess | ageGet() | rt_CMSIS.c:181 | 7 | | | | |
| 🔳 <no fu<="" td=""><td>unction Name></td><td></td><td></td><td></td><td></td><td></td><td></td></no> | unction Name> | | | | | | |
| | | | | | | | |
| Name | Value | Туре | Count | Size | Location | Access | |
| 🖃 🌒 a1 | 0x8030611C | osMessageQld | 1 | 32 | \$R4 | R/W | |
| – 🧬 a2 | 4294967295 | uint32_t | | 32 | \$R5 | R/W | |

| 5 Debug Cont 🔀 🍋 Project Expl 📕 Remote Syst | |
|---|-----|
| ∃ 🧠 🦎 💥 💥 🦓 🔍 ▼ 🖓 ▼ 🕨 🔍 🍮 . | 🖻 🚔 |
| \bigtriangledown | |
| 🔖 Big Snapshot connected | * |
| 🍇 swapper/1 #6 stopped (PID 0) | |
| Active Threads | |
| 🝇 swapper/1 #6 stopped (PID 0) | = |
| 🝇 swapper/2 #7 stopped (PID 0) | |
| 🝇 swapper/0 #8 stopped (PID 0) | |
| 🍇 swapper/3 #9 stopped (PID 0) | |
| h sh #10 stopped (PID 1355) | |
| 🍇 swapper/5 #11 stopped (PID 0) | |
| All Threads | |
| 🔆 CSAL disconnected | |
| 🔆 Generic-Snapshot-Juno disconnected | |
| 🔆 Generic-Snapshot-Snowball disconnected | - |
| Status: connected OS Support: Enabled | |
| 11 | |

Target connection configuration

- Hardware and models connection wizard
- Select device, board or FVP easily
 - Recently used list
 - Add new platform if device not listed
 - Creates a new configuration with the platform configuration editor
- Configuration wizard generates an initial debugger configuration based on device information.

| 📸 Hardware Connection — | | × |
|--|-------|---|
| Target Selection | Z | 3 |
| Select a target to debug | | A |
| Include uninstalled device packs | | |
| type filter text | | |
| Cortex-M Prototyping System (MPS2) Cortex-M0 Cortex-M Prototyping System (MPS2) Cortex-M0+ Cortex-M Prototyping System (MPS2) Cortex-M1 Cortex-M Prototyping System (MPS2) Cortex-M3 Cortex-M Prototyping System (MPS2) Cortex-M4 Cortex-M Prototyping System (MPS2) Cortex-M7 (SMM-M7) Cortex-M Prototyping System (MPS2) Cortex-M7 (SMM-M7CS) Cortex-M Prototyping System (MPS2+) Cortex-M2 IoT Cortex-M Prototyping System (MPS2+) Cortex-M3 DesignStart Cortex-M Prototyping System (MPS2+) Cortex-M3 IoT Cortex-M Prototyping System (MPS3) Cortex-M33 IoT EB-CT-R4F | | ^ |
| Add a new platform | | |
| Device: Cortex-M Prototyping System (MPS2) Cortex-M3 | | |
| Core(s): Cortex-M3 | | |
| Location: Configuration Database - configdb | | |
| No description available | | |
| ? < <u>B</u> ack <u>N</u> ext > <u>F</u> inish | Cance | : |



Debugger configuration with software packs

Device selection based on packs

🚼 C Project

Select Device ARM Cortex-M33 Device: Musca:Cortex-M33-0 CPU: Vendor: ARM Max. Clock: 50 MHz ARM.Musca_A1_BSP.2.0.0 4352 KB RAM, 512 KB ROM Pack: Memory: URL: http://www.keil.com/dd2/arm/musca EDI1-Search: musca Endian: Little-endian 🗸 🧳 ARM Musca SoC (TrustZone enabled 🗸 🔧 ARM Cortex M33 subsystem). 🗸 🔧 Musca The flag __DOMAIN_NS is used in the Musca:Cortex-M33-0 internal files to distinguish between Musca:Cortex-M33-1 secure and non-secure application. Please, use the flag as follow: -> secure application uses -D DOMAIN NS=0 -> non-secure application uses -D_DOMAIN_NS=1 ? < Back Next > Finish Cancel

 \Box \times

| 🚸 Blinky (Bulb Boa | rd) S6E1A12B | 0A Flash | | | | | |
|---------------------|--------------|--------------------------|----------------------|-------------|--------------------------|----------------|--------------------------|
| | | development board)_Targe | et 1 | | Connect and reset | | |
| · · | | 1 12 3 | - | | No reset O Pre-c | onnect reset (|) Hold reset and connect |
| | | | | | No reset will be perform | med when conn | ecting to the target. |
| | | | | | Reset control | | |
| Connection Settings | | Connection 🐟 Advanced 🭕 | Flash 🚸 OS Awareness | | | | _ |
| Connection Type | ULINKpr | Programming Algorithms | | | | | and reset the system. |
| Debug Port | SWD ~ | File | Region Start Address | Region Size | RAM Start Address | RAM Size | |
| Debug Port | 3000 ~ | S6E1A11X0A.FLM | 0x0000000 | 0×E000 | 0×2000000 | 0x1800 | |
| Connection Address | | | | | | | pt (.ds / .py) |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | ipt (.ds / .py) |

Advanced target connection, reset options

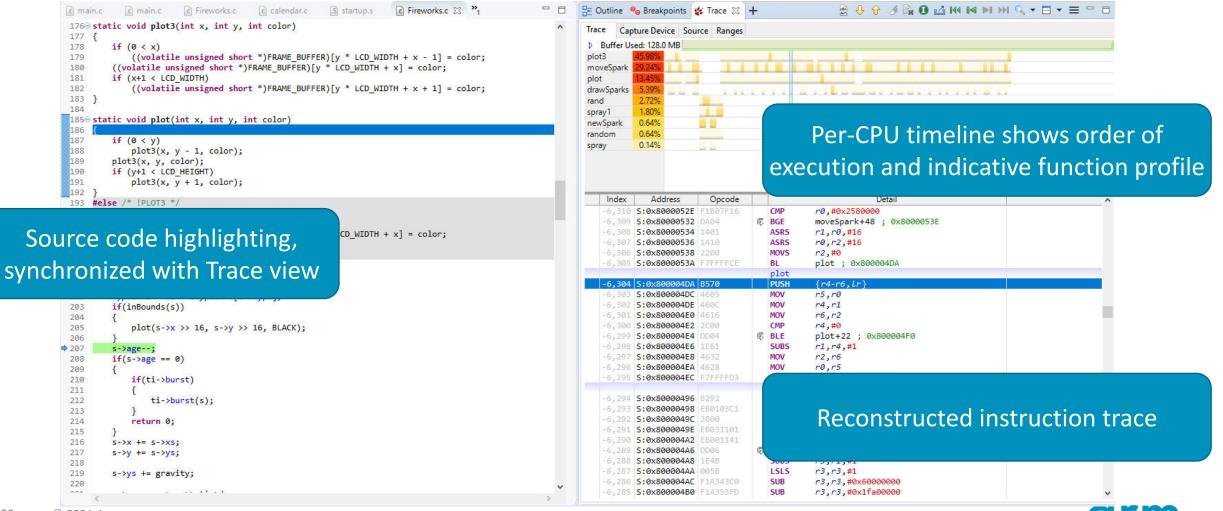
Flash programming with support for multiple flash regions

Trace – how it works

- CPU instruction execution history via ETM
 - Data load and store (certain CPUs)
 - Instrumented trace (ITM, STM)
- Arm trace data collected in
 - On-chip CoreSight trace buffer (ETB, ETF, ETR)
 - Off-chip debug probe via physical trace port on the target
- Rotating trace buffer fills in the background during a debug session
 - When a breakpoint is hit the content of the trace buffer is displayed

Trace view in Arm Debugger

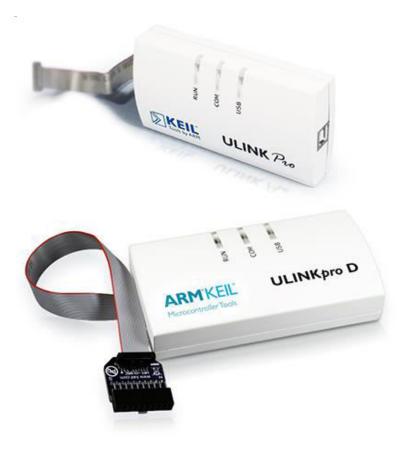
A non-intrusive 'flight recorder' for your system



Debug probes

Range of probes to match your requirements







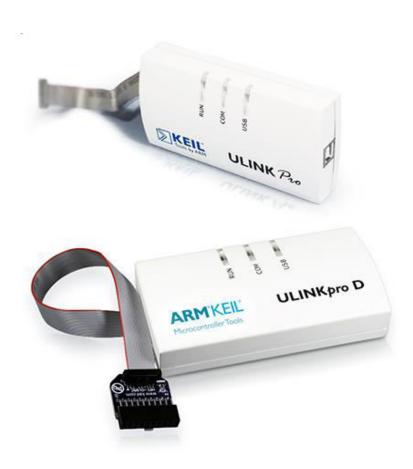
DSTREAM-ST family

Second-generation debug and trace probes

• New design featuring patented technology to address the needs of modern SoC's

| Capabilities | DSTREAM-ST | DSTREAM-PT | DSTREAM-HT |
|------------------------------|-----------------------------|-----------------------------|-----------------------------|
| JTAG / SWD | 180 / 125 MHz | 180 / 125 MHz | 180 / 125 MHz |
| Download speed | Up to 12MB/s | Up to 12MB/s | Up to 12MB/s |
| Maximum CoreSight components | 1022 | 1022 | 1022 |
| Parallel trace pins | 1 - 4 | 1 - 32 | 1 - 4 |
| High-speed serial trace | | | Yes |
| Maximum trace bandwidth | 1.6Gb/s | 19.2Gb/s | 60Gb/s |
| Host connectivity | USB 3.0
Gigabit Ethernet | USB 3.0
Gigabit Ethernet | USB 3.0
Gigabit Ethernet |
| On-probe trace store | Stream to host only | 8GB | 8GB |

ULINK family



Simplicity Just select a supported platform from the database and connect

Consistent debugging experience across probes

Cost Perfectly engineered to meet basic debug needs efficiency

Supported by both Keil and Arm Development Studio tools to protect your investment



Virtual platforms

+ + + + + + + + + + + + + + +

* * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * *

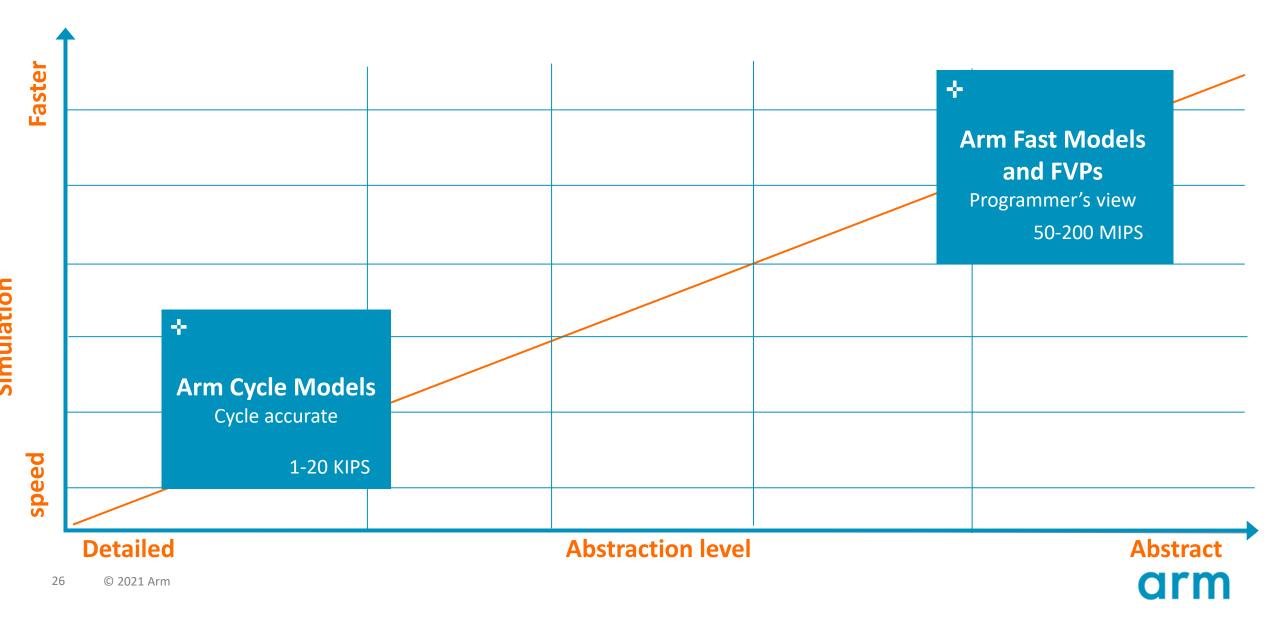
+ + + + + + + + + + + + + + +

* * * * * * * * * * * * * *

Virtual prototyping solutions

- Fast Models
 - Instruction accurate, suitable for software development and test
- Cycle Models
 - Timing accurate models, derived from actual RTL
- Fully validated models of Arm IP
- Collection of Fixed Virtual Platforms (FVP) provided with Development Studio
 - Complete system models based on Fast Model technology

Virtual prototype solutions



drm Arm Streamline performance

analyzer

+ + + + + + + + + + + + + + + +

* * * * * * * * * * * * * * * *

+ + + + + + + + + + + + +



Speed Up Your Code

• Find out where the CPU is spending the most time

Tune code for optimal cache usage

> 38.24% 30.37% 22.45% 6.75% 6.75% 4.09% 100.00% 100.00% 100.00% 100.00%

14.74% 5.57% 5.36% 0.04% 0.99% 0.38% 0.36% 0.01%

0.041 0.01% 0.68% 0.25% 0.22% 0.01%

3.72% 3.19% 0.16%

1 do_fracta

4 calculate 2 calccolumn_3 1 mkrealloc tab

2 calcline 3.

0 (# [kernel

calcline_32

mand cale

om.c:153

zoomd.c.31 calculate.h:5 zoomd.c:137 zoom.c:444 zoom.c:1210

Drill down to the Source Code

Break performance down by function

> View it alongside the disassembly



Arm Streamline

Performance Analyzer

OpenCL[™] Visualizer



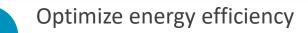
Visualization of OpenCL dependencies, helping you to balance resources between GPU and CPU better than ever



Mali GPU Support

■ Analyze and optimize Mali[™] GPU utilization

Monitor CPU and GPU cache usage



Monitor actual power consumption with

the Arm Energy Probe

Correlate software execution to actual

power consumption



Customize it for Your System

- Flexible architecture permits easy addition of new counters
- Open-source driver and daemon gives developers ultimate flexibility

Process to Success

| 🚽 Timeline 💋 C | all Paths 🚳 | Functions 🗟 Cod | de < Call Graph | 🗏 Stack 💞 Log |
|------------------------------------|-------------|-----------------|-----------------|------------------------|
| 0 | F | | Q | Sma. V Mr. mn. a |
| 7.4s 7.4 | | | 7.65s 7.7s | 7,75s 7,8s 7,85s |
| CPU Activity | S 1007 | | | 9% arg.
0% arg. |
| Instruction Executed | ф 🖬 ЗМ
6 | | 013.8 | 53,688 |
| Clock Frequency Cycles | 7001 | MHz | 0175 | 690 MHz
09 MHz avg. |
| Branch
Mispredicted | (5) | ¢ | © 203, | 326 |
| Cache | (3) | | • 199, | 801 |
| 1] (<i>f</i>) | 4 | | | |
| [die] | | | | |
| • (kernel) | | | | |
| (xaos #3060) | | | | - |
| [gatord #3086] | | | | |
| [Xorg #2174] | | | | |

| 🔜 Tim | eline 💋 🤇 | Call Paths | 🚯 Fun | ctions 🔜 Code < Call Graph 🗏 Stack 🍕 | 🗲 Log 💡 |
|-------|-----------|------------|----------|--------------------------------------|-------------------------|
| ? | % | Σ | | | |
| Self | Process | Total 👻 | Stack | Process/Thread/Function Name | L |
| 0.00% | 100.00% | 81.06% | 0 | | - |
| 0.00% | 27.10% | 21.97% | 0 | 🖶 [thread #863] | - |
| 0.00% | 22.43% | 18.18% | 0 | pthread_initialize_minimal | libpthr |
| 0.00% | 22.43% | 18.18% | 144 | i control_routine | thread. |
| 0.00% | 22.43% | 18.18% | 288 | processqueue | zoom.c |
| 0.93% | 13.08% | 10.61% | 464 | calcline_32 | zoomd |
| 0.93% | 12.15% | 9.85% | 528 | 😑 calculate | calcula |
| 8.41% | 8.41% | | | - mand_peri | docalc. |
| 2.80% | 2.80% | 2.27% | 592 | -mand_calc | docalc. |
| 0.00% | 0.00% | 0.00% | 592 | calculate | calcula |
| 1.87% | 9.35% | 7.58% | 464 | calccolumn_32 | zoomd |
| 0.93% | 7.48% | 6.06% | 528 | 🖨 calculate | calcula |
| 4.67% | 4.67% | 3.79% | 592 | - mand_peri | |
| 0.93% | 0.93% | 0.76% | 592 | - calculate | calcula |
| 0.93% | 0.93% | 0.76% | 592 | mand_calc | docalc. |
| 0.00% | 0.00% | 0.00% | 288 | pthread_mutex_lock | libpthr |
| 0.00% | 0.00% | 0.00% | 272 | - bfill32 | btraced |
| 0.00% | 0.00% | 0.00% | 272 | - dosymetry2_32 | zoomd |
| 0.00% | 0.00% | 0.00% | 288 | 🕀 filly | zoom.c |
| 0.00% | 0.00% | 0.00% | 320 | - mkrealloc_table | zoom.c |
| 0.00% | 0.00% | 0.00% | 288 | moveoldpoints | zoom.c |
| 0.00% | 0.00% | 0.00% | 144 | pthread_mutex_timedlock | libpthre |
| 0.00% | 0.00% | 0.00% | 288 | i queue32 | btraced |
| 4.67% | 4.67% | 3.79% | 0 | -[libc-2.11.1.so] | <anony< td=""></anony<> |
| 0.00% | 0.00% | 0.00% | 0 | pthread_cond_wait | libpthre |
| 0.00% | 0.00% | 0.00% | 0 | - oft (vaor) | |

| 🔜 Time | eline 💋 Cal | l Paths 🔞 Functions 🗟 | Code | - | Call Graph | | | | | |
|--------|-------------|--------------------------------------|------------------------|--------------|-----------------------------|--|--|--|--|--|
| 0 | % Σ | | | | _ | | | | | |
| Self 👻 | Instances | Function Name | | | Location | | | | | |
| 54.25% | 1 | [idle] | dle] | | | | | | | |
| 13.89% | 77 | [libdvm.so] | ibdvm.so] | | | | | | | |
| 13.78% | 1 | 1 [kernel] <unknow< td=""></unknow<> | | | | | | | | |
| 3.50% | 74 | [libc.so] | | | <unknown< td=""></unknown<> | | | | | |
| 2.43% | 16 | [libskia.so] | - | h
bura da | Kunknown | | | | | |
| 1.57% | 9 | [libz.so] | Call Paths | | | | | | | |
| 1.22% | 17 | [libMali.so] | t Process/Th | | | | | | | |
| 1.09% | 33 | [dev/ashmem/dalvik-jit-c | | | | | | | | |
| 1.09% | 1 | [libGLESv1_CM_mali.so] | t in Call Patl | | | | | | | |
| 0.89% | 37 | [libcutils.so] | t in Code | | | | | | | |
| 0.87% | 42 | [libutils.so] | [libutils.so] < Select | | | | | | | |
| 0.78% | 2 | [gator] | E | Selec | t in Stack | | | | | |
| 0.63% | 5 | [libhwui.so] | - | 1000 | | | | | | |
| 0.54% | 6 | [libGLESv2_mali.so] | | | <unknown< td=""></unknown<> | | | | | |
| 0.46% | 38 | [libbinder.so] | | | <unknown< td=""></unknown<> | | | | | |
| 0.44% | 2 | [libcrypto.so] | | | <unknown< td=""></unknown<> | | | | | |

| ., | Timeline | ø | Call | Paths | G | Functions | ; | Code | < Call Graph 🗉 |
|----|----------|---|------|--------|--------|-------------|-----|----------|---------------------|
| ?) | 111 | % | Σ | c) #4 | 1 | | | | |
| S | amples | I | Line | Source | e File | e: C:/Users | /qu | imar01/ | DS-5/Workspaces/5.4 |
| | | | | | | | | | gned int threa |
| | | | 108 | { | | | | | - |
| | | | 109 | | uns | igned : | int | i; | |
| | | | 110 | | | | | | |
| | | | 111 | 1 | pri | ntf("T | hre | ad %d | started accur |
| | 8.62% | | 112 | | for | (i=0; | i< | loops | ; i++) |
| | | | 113 | | { | | | - | |
| | 74.14% | | 114 | | | accum | = | accum | + step; |
| | 17.24% | | 115 | | | if (i | ==1 | .oops/ | 2) |
| | | | 116 | | | { | | | |
| | | | 117 | | | p | rin | tf("T | hread %d half |
| _ | | | 118 | | | 1 | | | |
| | | | | | | | | | |
| S | amples | I | A | ddress | | Opcode | : | Disasser | nbly |
| | 29.31% | | 0x0 | 00089 | oc | | 05 | MOV | r0, r5 |
| | 10.34% | | 0x00 | 00089 | 10 | | 07 | MOV | r1, r7 |
| | | | 0x00 | 00089 | 14 | | | BL | _addsf3 ; |
| | 34.48% | | 0x0 | 00089 | 18 | | 00 | MOV | r5,r0 |
| | | | 0x00 | 00089 | 1C | E15800 | 04 | CMP | r8,r4 |
| | 17.24% | | | | | 1A0000 | | | 0x00008930 |
| | | | 0x00 | 00089 | 24 | E1A000 | 09 | MOV | r0, r9 |
| | | | 0x0(| 00089 | 28 | E1A010 | DA | MOV | r1, r10 |
| | | | | | | EBFFFF: | | | {pc}-0x348 |
| | | | | | | E28440 | | | r4,r4,#1 |
| | 8.62% | | 0x00 | 00089 | 34 | E15600 | 04 | CMP | r6,r4 |
| | 0.020 | | | | | | | | |

| 0 | | | | Log Entries:
Delta: | | | | |
|--------------|------------|---------------------|------------|------------------------|-----|------------------|---|--|
| Message reg | | | Core regex | Where regex | | | | |
| When | Delta | Message | Core | | | ere | | |
| 00:37.428088 | +00.000000 | Red 0 | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.429088 | +00.001000 | Blue 00001 | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 0:37.430087 | +00.000999 | Green 2 | 0 | (annotate #181 | 04] | [[thread #18104 |] | |
| 00:37.431087 | | | | | | [thread #18104 | | |
| | | Yellow 4.000000e+00 | | | | [thread #18104 | | |
| 0:37.433088 | +00.001001 | Cyan 5 | | | | [thread #18104 | | |
| 00:37.434087 | | | | | | [thread #18104 | | |
| | | LtGray 00007 | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| | | DkGray 00008 | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.437091 | | | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.438091 | +00.001000 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.439093 | +00.001002 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.440091 | +00.000998 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 00:37.441091 | +00.001000 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.442090 | +00.000999 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.443089 | +00.000999 | Repeat | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 00:37.444090 | +00.001001 | Red 10 | 0 | [annotate #181 | 04] | [thread #18104 |] | |
| 0:37.454090 | | | 0 | (annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.464090 | +00.010000 | Green c | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.474088 | +00.009998 | Purple 13.00 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.484093 | +00.010005 | Yellow 1.400000e+01 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 00:37.494090 | +00.009997 | Cyan 17 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.504090 | +00.010000 | White 00016 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 00:37.514088 | +00.009998 | LtGray 00017 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.524087 | +00.009999 | DkGray 00018 | 0 | [annotate #181 | 04] | [thread = 18104 | 3 | |
| 0:37.534087 | +00.010000 | Black 00019 | 0 | [annotate #181 | 04] | [thread #18104 | 1 | |
| 0:37.544088 | +00.010001 | Repeat | | | | [thread #18104 | | |
| 0:37.554087 | +00.009999 | Repeat | | | | [thread #18104 | | |
| 0:37.564087 | +00.010000 | Repeat | | | | [thread = 18104 | | |
| 0-17 574097 | +00.010000 | Repeat | | | | [[thread #18104 | | |

Timeline

Visualization of system performance metrics, software profile and system events over time

Call Paths

Hierarchical profile table, aggregating samples per process, thread, and function call chain

Functions

Flat software profile table, listing shared libraries and function hotspots

Code

Source and instruction level profile. Color coded source code lines for easy identification

Log

Chronologic list of text and graphic user annotations sent to Streamline, offering flexible filtering

We invite you to try it out for yourself

For more information see arm.com/development-studio



You're One Step Away from Starting Your 30 Day Free Trial

Every feature, no credit card payment required

Try for yourself every component of Arm Development Studio Gold for free for 30 days – build, simulate, debug and optimize prepackaged examples or your own code to experience hands-on how Arm's flagship toolchain helps you achieve your design goals more efficiently.

Get started today.



