COMPUTER ARCHITECTURE

CE, Computer Engineering degree

Antonio Prete

a.prete@ing.unipi.it

- Measuring Performance;
- Benchmarks;
- Principles of Computer Design.

Measuring Performance

- Typical performance metrics:
  - Response time
  - Throughput
- Execution time
  - Wall clock time: includes all system overheads
  - CPU time: only computation time
- Speedup of X relative to Y
  - Execution time$_Y$ / Execution time$_X$
- Benchmarks
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)
  - Synthetic benchmarks (e.g. Dhrystone)
  - Benchmark suites (e.g. SPEC06fp, TPC-C)

Performance metrics

We often want to relate the performance of two different computers, say, X and Y.

- Response time (t)
- Throughput (1/t)

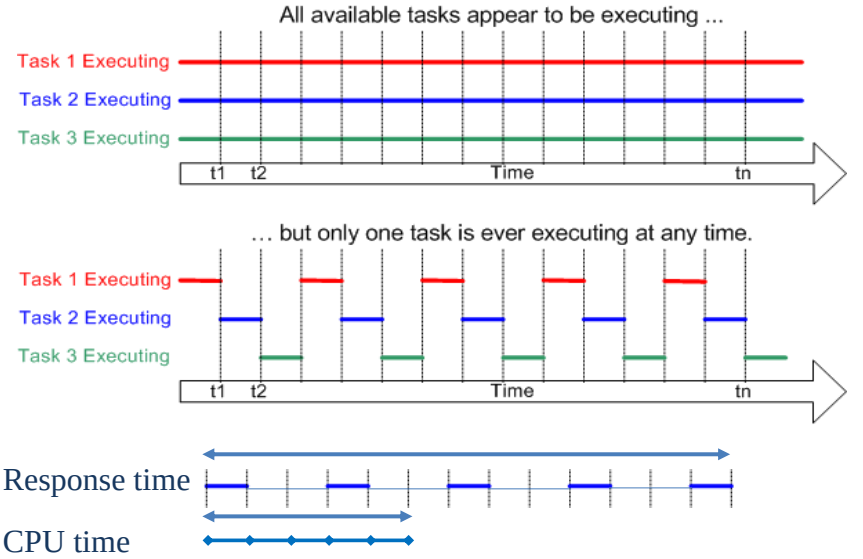
"X is 4 times as fast as Y" will mean:

$$\frac{\text{Execution time}_Y}{\text{Execution time}_X} = 4$$

"the throughput of X is 3 times as fast as Y" signifies here that the number of tasks completed per unit time on computer X is 3 times the number completed on Y.

$$\frac{\text{Execution time}_Y}{\text{Execution time}_X} = \frac{\frac{1}{\text{Performance}_Y}}{\frac{1}{\text{Performance}_X}} = \frac{\text{Performance}_X}{\text{Performance}_Y} = 3$$

Response time vs CPU time



5

5

Measuring Performance (I)

The best choice of benchmarks to measure (to compare) performance is real applications.

Attempts at running programs that are much simpler than a real application have led to performance pitfalls.

Benchmarks

- Kernels (e.g. matrix multiply)
- Toy programs (e.g. sorting)
- Synthetic benchmarks (e.g. Dhrystone)
- Benchmark suites (e.g. EEMBC, SPEC06fp, TPC-C)
  - www.spec.org
  - www.eembc.org
  - www.TCP.org

6

6

Measuring Performance (II)

Compiler writer and architect ***can conspire*** to make the computer appear faster on these stand-in programs than on real applications.

Another issue is the ***conditions*** under which the benchmarks are run.

- One way to improve the performance of a benchmark has been with benchmark-specific compiler flags; these flags often caused transformations that would be illegal on many programs or would slow down performance on others.

Measuring Performance (III)

Source code modifications

1. No source code modifications are allowed.
2. Source code modifications are allowed but are essentially impossible.

For example, database benchmarks rely on standard database programs that are tens of millions of lines of code.

The database companies are highly unlikely to make changes to enhance the performance for one particular computer.

3. Source modifications are allowed, as long as the altered version produces the same output.

Benchmark tests value

- One problem that affects manufacturers' benchmark tests is that test programs could be chosen or constructed in such a way as to privilege the characteristics of their products.
- In 2014 Intel closed a ten-year class action, paying off some of the users who between November 2000 and June 2002 bought a first generation Pentium 4 (Willamette) instead of an AMD Athlon, as the benchmark tests returned better values for the Pentium 4, in how much the tests were based on operations in which the Pentiums were better, avoiding those in which the Athlons would have excelled.

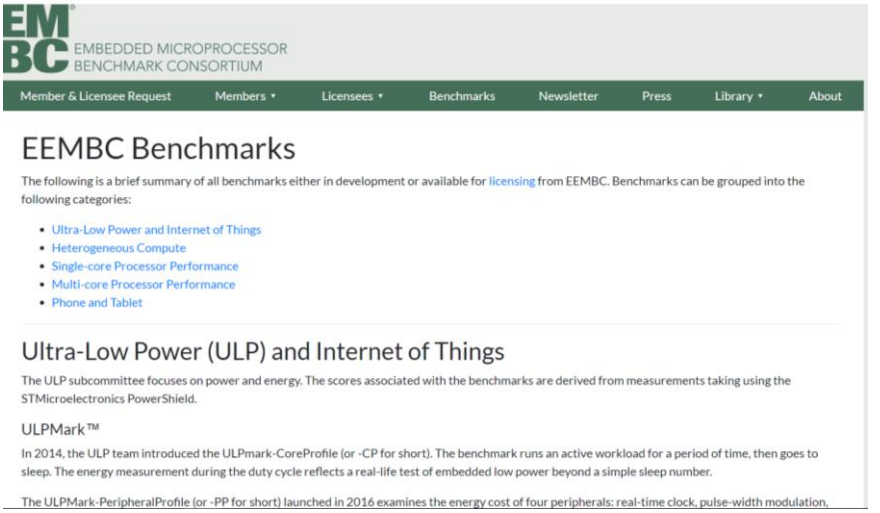
In the end the score was calculated correctly, but it was not necessarily true.

Electronic Design News Embedded Microprocessor Benchmark Consortium (EEMBC “embassy”) benchmarks

It is a set of 41 kernels used to predict performance of different embedded applications: automotive/industrial, consumer, networking, office automation, and telecommunications.

EEMBC reports unmodified performance and “full fury” performance, where almost anything goes.

EEMBC ("embassy"): the web site



Ultra-Low Power (ULP) and Internet of Things focus on power and energy

The scores associated with the benchmarks are derived from measurements taking using the STMicroelectronics PowerShield.

- ULPMark™
  - The benchmark **runs an active workload** for a period of time, then goes to **sleep**. The energy measurement during the **duty cycle** reflects a real-life test of embedded low power beyond a simple sleep number.
- ULPMark-PeripheralProfile
  - It examines the energy cost of four peripherals: **real-time clock**, pulse-width **modulation**, **analog-to-digital conversion**, and **SPI.** communication.
- ULPMark-CoreMark
  - It measures the energy of CoreMark in a consistent environment. The ULPMark-CM score is proportional to the number of CoreMark iterations a device can execute per milli-Joule.

Ultra-Low Power (ULP) and Internet of Things focus on power and energy

IoTMark™

- IoTMark builds on ULPMark by adding a **sensor emulation module** (the IO Manager) and a radio **gateway emulator** (the Radio Manager).
- The execution profile incorporates **the types of behavior an IoT edge node** would perform.
  - The first benchmark in this series, IoTMark-BLE, uses a Bluetooth Low Energy (BLE) radio as the gateway and an I2C device as the sensor.

SecureMark™

- Security comes at a cost, both in programming complexity and energy. As IoT secure becomes even more important, designers must assess the energy cost of the design. SecureMark provides **security-specific profiles to assist in this analysis**.

13

EEMBC, Microprocessor Benchmark Suites

PROCESSOR

AutoBench™ Automotive, industrial, and general-purpose applications.

ConsumerBench™ - Digital cameras, printers, and other **digital imaging systems**.

CoreMark® - For quick comparison of processor and microcontroller core functionality.

DENBench™ - Digital entertainment products such as smartphones, MP3 players, digital cameras, TV set-top boxes, and in-car entertainment systems.

EnergyBench™ - Power and energy performance with insights to power budget costs.

FPBench™ - **Floating-point performance** in graphics, audio, motor control, and other high-end processing tasks.

MultiBench™ - Multicore architectures, memory bottlenecks, OS thread scheduling, and efficiency of synchronization.

Networking - Moving and analyzing packets in networking applications.

OABench™ - Office Automation tasks in printers, plotters, and other systems that handle text and image processing tasks.

TeleBench™ - Telecommunications processors in modem, xDSL, and related fixed-telecom applications.

14

Phone and Tablet

BrowsingBench™

- Measure browser performance with BrowsingBench, a collection of webpages loaded using a local Nginx server over a wired LAN connection.

AndEBench™-Pro

- Available for free in the Android Play Store, AndeBench is an industry-accepted method of evaluating Android platform performance.

15

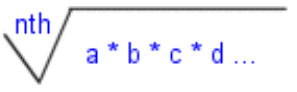
How to calculate the score

The score reported for each device is a single-number figure of merit calculated by taking the geometric mean of the individual AutoBench scores and dividing by **307.455**.

This normalization factor (307.455) is derived from the lowest score in this category on December 5, 2000.

Scores for each of the individual benchmarks within this suite allow designers to aggregate the benchmarks to suit specific application requirements.

To calculate a **geometric mean**, multiply all the results of the tests together and take the nth root of the product, where n equals the number of tests.



16

EEMBC, Microprocessor Benchmark Suites
Services

EEMBC Technology Center (ETC)

Services include **porting and benchmark execution, performance analysis, and preparation of platforms for benchmark score certification**.

These services allow any company to make EEMBC benchmark testing an integral part of its product development and release process - without tying up internal engineering resources.

Scores for ULPMark-CP and ULPMark-PP

Clear	Hardware	Vendor Score	Cert.	Core	Compiler	Core Profile (3.0 V)	Core Profile (User)	Periph. Profile (3.0 V)	Periph. Profile (User)	Date↓
	STMicroelectronics STM32L412 Rev A	✓	✓	Cortex-M4	IAR C/C++ Compiler for ARM 8.20.2	247	447 1.8V	94.0	167 1.8V	2018-10-17
	STMicroelectronics STM32L552 Rev1	✓		Cortex-M33	IAR C/C++ Compiler for ARM 8.20.1	267	402 1.8V	33.5	59.5 1.8V	2018-10-15
	Analog Devices ADuCM4050 Rev 0.1	✓	✓	Cortex M4F	IAR EWARM 8.20.1.14188			24.3		2018-07-04
	Microchip Technology ATSAML11E16A rev B	✓	✓	ARM Cortex-M23	IAR C/C++ Compiler for ARM 8.22.1.15669	282	400 1.8V			2018-04-10
	Microchip Technology ATSAML10E16A rev B	✓	✓	ARM Cortex-M23	IAR C/C++ Compiler for ARM 8.22.1.15669	281	405 1.8V			2018-04-10
	Analog Devices ADuCM4050 Rev 0.1	✓	✓	Cortex M4F	IAR EWARM 8.20.1.14188	189				2018-03-20
	Analog Devices ADuCM302x Rev1.0			Cortex-M3	ARM GCC 7-2017-q4-major			3.43	3.47 1.8V	2018-03-12
	Silicon Labs EFM32PG1B200F256 + 32KB SRAM retention			Cortex-M4	Silicon Labs Simplicity Studio v3		135 1.8V			2018-02-11
	Silicon Labs EFM32PG1B200F256			Cortex-M4	Silicon Labs Simplicity Studio v3	106	144 1.8V			2018-02-11
	Silicon Labs EFM32HG322F64			Cortex-M0+	Silicon Labs Simplicity Studio v3	101	157 2.0V	4.84	8.03 2.0V	2018-02-09
	Silicon Labs EFM32LG990F256			Cortex-M3	Silicon Labs Simplicity Studio v3	74.2	114 2.0V	35.9	57.2 2.0V	2018-02-09
	Texas Instruments MSP432P401R Rev. C + BOD/SVS + 64K SRAM			Cortex-M4	IAR EWARM v7.50.3	150	208 1.8V	7.08	12.4 1.8V	2018-02-09

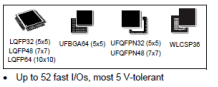
STM32L412xx

Ultra-low-power ARM® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 128KB Flash, 40KB SRAM, analog, ext. SMPS

Data brief

Features

- Ultra-low-power with FlexPowerControl
  - 1.71 V to 3.6 V power supply
  - -40 °C to 85/125 °C temperature range
  - 145 nA in V_{BAT} mode: supply for RTC and 32x32-bit backup registers
  - 8 nA Shutdown mode (4 wakeup pins)
  - 28 nA Standby mode (4 wakeup pins)
  - 280 nA Standby mode with RTC
  - 1.0 µA Stop 2 mode, 1.28 µA with RTC
  - 84 µA/MHz run mode (LDO Mode)
  - 36 µA/MHz run mode (@3.3 V SMPS Mode)
  - Batch acquisition mode (BAM)
  - 4 µs wakeup from Stop mode
  - Brown out reset (BOR)
  - Interconnect matrix
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS and DSP instructions
- Performance benchmark
  - 1.25 DMIPS/MHz (Drystone 2.1)
  - 273.55 CoreMark® (3.42 CoreMark/MHz @ 80 MHz)
- Energy benchmark
  - 174.5 ULPBench® Core Profile score
- Clock Sources
  - 4 to 48 MHz crystal oscillator
  - 32 kHz crystal oscillator for RTC (LSE)
  - Internal 16 MHz factory-trimmed RC (±1%)
  - Internal low-power 32 kHz RC (±5%)
  - Internal multispeed 100 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than ±0.25 % accuracy)
  - Internal 48 MHz with clock recovery
  - PLL for system clock and ADC



- Up to 52 fast I/Os, most 5 V-tolerant
- RTC with HW calendar, alarms and calibration
- Up to 12 capacitive sensing channels: support touchkey, linear and rotary touch sensors
- 10x timers: 1x 16-bit advanced motor-control, 1x 32-bit and 2x 16-bit general purpose, 1x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
- Memories
  - 128 KB single bank Flash, proprietary code readout protection
  - 40 KB of SRAM including 8 KB with hardware parity check
  - Quad SPI memory interface with XIP capability
- Rich analog peripherals (independent supply)
  - 2x 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
  - 2x operational amplifier with built-in PGA
  - 1x ultra-low-power comparator
  - Accurate 2.5 V or 2.048 V reference voltage buffered output
- 12x communication interfaces
  - USB 2.0 full-speed crystal less solution with LPM and BCD
  - 3x I2C FM+(1 Mbit/s), SMBus/PMBus
  - 3x USARTs (ISO 7816, LIN, IrDA, modem)
  - 1x LPUART (Stop 2 wake-up)
  - 2x SPIs (3x SPIs with the Quad SPI)
  - IRTIM (Infrared interface)
- 14-channel DMA controller
- True random number generator

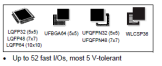
STM32L412xx

Ultra-low-power ARM® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 128KB Flash, 40KB SRAM, analog, ext. SMPS

Data brief

Features

- Ultra-low-power with FlexPowerControl
  - 1.71 V to 3.6 V power supply
  - -40 °C to 85/125 °C temperature range
  - 145 nA in V_{BAT} mode: supply for RTC and 32x32-bit backup registers
  - 8 nA Shutdown mode (4 wakeup pins)
  - 28 nA Standby mode (4 wakeup pins)
  - 280 nA Standby mode with RTC
  - 1.0 µA Stop 2 mode, 1.28 µA with RTC
  - 84 µA/MHz run mode (LDO Mode)
  - 36 µA/MHz run mode (@3.3 V SMPS Mode)
  - Batch acquisition mode (BAM)
  - 4 µs wakeup from Stop mode
  - Brown out reset (BOR)
  - Interconnect matrix
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS and DSP instructions
- Performance benchmark
  - 1.25 DMIPS/MHz (Drystone 2.1)
  - 273.55 CoreMark® (3.42 CoreMark/MHz @ 80 MHz)
- Energy benchmark
  - 174.5 ULPBench® Core Profile score
- Clock Sources
  - 4 to 48 MHz crystal oscillator
  - 32 kHz crystal oscillator for RTC (LSE)
  - Internal 16 MHz factory-trimmed RC (±1%)
  - Internal low-power 32 kHz RC (±5%)
  - Internal multispeed 100 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than ±0.25 % accuracy)
  - Internal 48 MHz with clock recovery
  - PLL for system clock and ADC

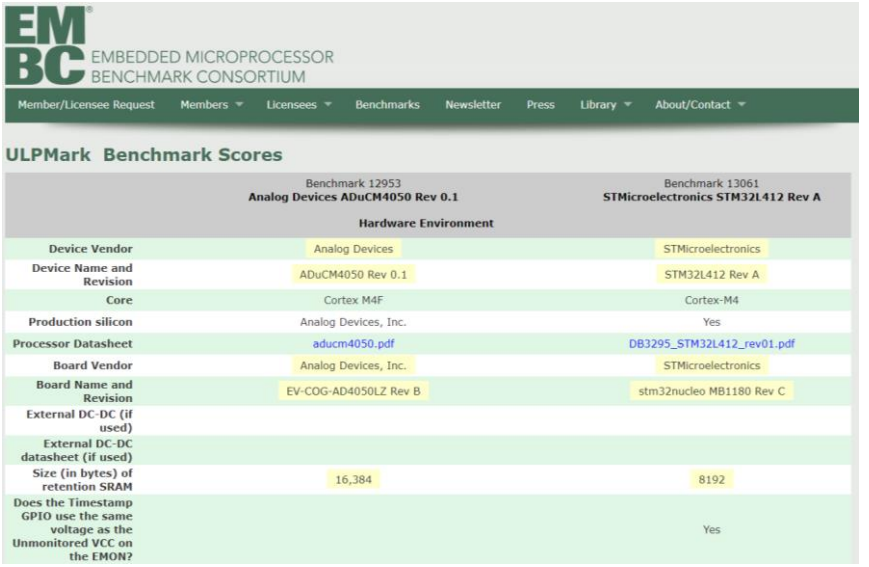


- Up to 52 fast I/Os, most 5 V-tolerant
- RTC with HW calendar, alarms and calibration
- Up to 12 capacitive sensing channels: support touchkey, linear and rotary touch sensors
- 10x timers: 1x 16-bit advanced motor-control, 1x 32-bit and 2x 16-bit general purpose, 1x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
- Memories
  - 128 KB single bank Flash, proprietary code readout protection
  - 40 KB of SRAM including 8 KB with hardware parity check
  - Quad SPI memory interface with XIP capability
- Rich analog peripherals (independent supply)
  - 2x 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
  - 2x operational amplifier with built-in PGA
  - 1x ultra-low-power comparator
  - Accurate 2.5 V or 2.048 V reference voltage buffered output
- 12x communication interfaces
  - USB 2.0 full-speed crystal less solution with LPM and BCD
  - 3x I2C FM+(1 Mbit/s), SMBus/PMBus
  - 3x USARTs (ISO 7816, LIN, IrDA, modem)
  - 1x LPUART (Stop 2 wake-up)
  - 2x SPIs (3x SPIs with the Quad SPI)
  - IRTIM (Infrared interface)
- 14-channel DMA controller
- True random number generator

LQFP64 (10x10)

- Up to 52 fast I/Os, most 5 V-tolerant
- RTC with HW calendar, alarms and calibration
- Up to 12 capacitive sensing channels: support touchkey, linear and rotary touch sensors
- 10x timers: 1x 16-bit advanced motor-control, 1x 32-bit and 2x 16-bit general purpose, 1x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
- Memories
  - 128 KB single bank Flash, proprietary code readout protection
  - 40 KB of SRAM including 8 KB with hardware parity check
  - Quad SPI memory interface with XIP capability
- Rich analog peripherals (independent supply)
  - 2x 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
  - 2x operational amplifier with built-in PGA
  - 1x ultra-low-power comparator
  - Accurate 2.5 V or 2.048 V reference voltage buffered output
- 12x communication interfaces
  - USB 2.0 full-speed crystal less solution with LPM and BCD
  - 3x I2C FM+(1 Mbit/s), SMBus/PMBus
  - 3x USARTs (ISO 7816, LIN, IrDA, modem)
  - 1x LPUART (Stop 2 wake-up)
  - 2x SPIs (3x SPIs with the Quad SPI)
  - IRTIM (Infrared interface)
- 14-channel DMA controller
- True random number generator

An example of evaluation report Ultra-Low-Power device (1)



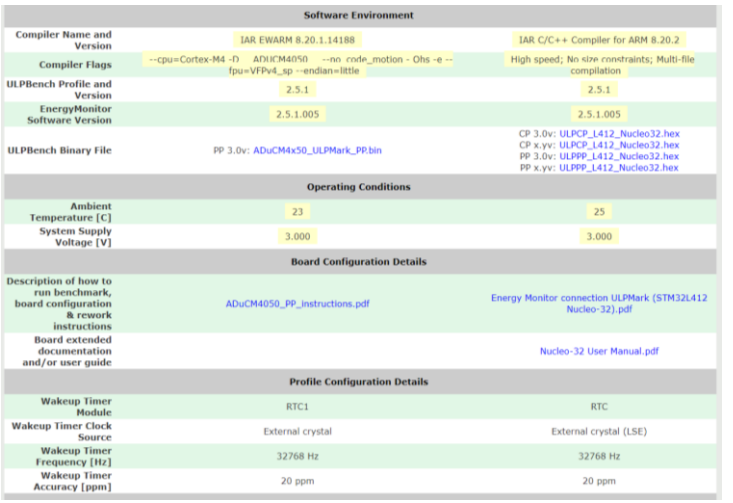
EMBC EMBEDDED MICROPROCESSOR BENCHMARK CONSORTIUM

Member/Licensee Request | Members | Licensees | Benchmarks | Newsletter | Press | Library | About/Contact

ULPMark Benchmark Scores

	Benchmark 12953 **Analog Devices ADuCM4050 Rev 0.1**	Benchmark 13061 **STMicroelectronics STM32L412 Rev A**
Hardware Environment		
Device Vendor	Analog Devices	STMicroelectronics
Device Name and Revision	ADuCM4050 Rev 0.1	STM32L412 Rev A
Core	Cortex M4F	Cortex-M4
Production silicon	Analog Devices, Inc.	Yes
Processor Datasheet	aducm4050.pdf	DB3295_STM32L412_rev01.pdf
Board Vendor	Analog Devices, Inc.	STMicroelectronics
Board Name and Revision	EV-COG-AD4050LZ Rev B	stm32nucleo MB1180 Rev C
External DC-DC (if used)		
External DC-DC datasheet (if used)		
Size (in bytes) of retention SRAM	16,384	8192
Does the Timestamp GPIO use the same voltage as the Unmonitored VCC on the EMON?		Yes

An example of evaluation report Ultra-Low-Power device (2)



Software Environment		
Compiler Name and Version	IAR EWARM 8.20.1.14188	IAR C/C++ Compiler for ARM 8.20.2
Compiler Flags	--cpu=Cortex-M4 -D ADUCM4050 --no_code_motion - Ohs -e --fpu=VFPv4_sp --endian=little	High speed; No size constraints; Multi-file compilation
ULPBench Profile and Version	2.5.1	2.5.1
EnergyMonitor Software Version	2.5.1.005	2.5.1.005
ULPBench Binary File	PP 3.0v: ADuCM4x50_ULPMark_PP.bin	CP 3.0v: ULPCP_L412_Nucleo32.hex CP x.yv: ULPCP_L412_Nucleo32.hex PP 3.0v: ULPPP_L412_Nucleo32.hex PP x.yv: ULPPP_L412_Nucleo32.hex
Operating Conditions		
Ambient Temperature [C]	23	25
System Supply Voltage [V]	3.000	3.000
Board Configuration Details		
Description of how to run benchmark, board configuration & rework instructions	ADuCM4050_PP_instructions.pdf	Energy Monitor connection ULPMark (STM32L412 Nucleo-32).pdf
Board extended documentation and/or user guide		Nucleo-32 User Manual.pdf
Profile Configuration Details		
Wakeup Timer Module	RTC1	RTC
Wakeup Timer Clock Source	External crystal	External crystal (LSE)
Wakeup Timer Frequency [Hz]	32768 Hz	32768 Hz
Wakeup Timer Accuracy [ppm]	20 ppm	20 ppm

An example of evaluation report Ultra-Low-Power device (2)

Benchmark Scores		
ULPMark-CP (3.0v)	n/a	**247.00**
ULPMark-CP (x.yv)	n/a	**447.00** (1.80v)
ULPMark-PP (3.0v)	**24.30**	**94.00**
Configuration 1	73.6	12.1
Configuration 2	95.8	15.6
Configuration 3	8.14	3.37
Configuration 4	13.1	4.94
Configuration 5	13.7	5.15
Configuration 6	13.7	5.18
Configuration 7	13.8	5.17
Configuration 8	13.8	5.18
Configuration 9	161	46.9
Configuration 10	4.94	2.8
ULPMark-PP (x.yv)	n/a	**167.00** (1.80v)
Configuration 1		6.98
Configuration 2		8.38
Configuration 3		1.64
Configuration 4		2.38
Configuration 5		2.52
Configuration 6		2.52
Configuration 7		2.52
Configuration 8		2.53
Configuration 9		29
Configuration 10		1.41

Highlighted Fields indicate Certified Data

An example of evaluation report Ultra-Low-Power device (2)

(in Median value)

Device	Device Score	CoreMark-Pro (Base)	CPU	GPU	Clock (MHz)
Xiaomi Tech MI NOTE PRO (MI NOTE PRO)	7747	2158	Qualcomm Snapdragon 810 MSM8994 v2.1	Qualcomm Adreno 430	1500
Huawei P10 Plus Premium Edition (VKY-AL00)	19812	6181	HiSilicon Honor KIRIN960 Hi3660	ARM Mali-G71	
Xiaomi Tech Redmi 3 (Redmi 3)	4961	2020	Qualcomm Snapdragon 410 MSM8916		
Xiaomi Tech MI 5s Plus (MI 5s Plus)	13602	4939	Qualcomm Snapdragon 820 MSM8996	Qualcomm Adreno 530	
Huawei Mate 9 (MHA-AL00)	16597	5453	HiSilicon Honor KIRIN960 Hi3660	ARM Mali-G71	
Xiaomi Tech Redmi Note 3 (Redmi Note 3)	9064	3634	Qualcomm Snapdragon 618 MSM8956	Qualcomm Adreno 510	1400
Motorola Moto Z (XT1650-05)	16099	4683	Qualcomm Snapdragon 820 MSM8996	Qualcomm Adreno 530	
Meizu PRO 6 Plus (PRO 6 Plus)	12405	4555	Samsung Exynos 8 Octa 8890	ARM Mali-T880	
Xiaomi Tech Mi5 (MI 5)	19508	4168	Qualcomm Snapdragon 820 MSM8996	Qualcomm Adreno 530	
Texas Instruments jacinto6evm (jacinto6evm)	4739	1871	Texas Instruments OMAP5		1500
Xiaomi Tech MI 5s (MI 5s)	9599	4266	Qualcomm Snapdragon 820 MSM8996	Qualcomm Adreno 530	
Huawei Honor 4A (SCL-AL00)	2585	797	Qualcomm Snapdragon 210 MSM8909	Qualcomm Adreno 304	1100
Huawei Mate 9 (MHA-L29)	18566	6154	HiSilicon Honor KIRIN960 Hi3660	ARM Mali-G71	
Huawei P10 Lite (WAS-LX1A)	7659	3157	HiSilicon Honor KIRIN658	ARM Mali-T830	

A report (I)
Freescale i.MX31-532MHz

EEMBC

AutoBench 1.1 Production Silicon Benchmark Scores

All Benchmark Scores are Certified by EEMBC to ensure credibility

Export Report to Excel

	Freescale i.MX31-532MHz
Certification Report	View Certification Report to see complete certification data
Type of Platform	Hardware/Production Silicon
Type of Certification	Out-of-the-box
Certification Date	01/21/08
Benchmark Notes	
Hardware Type	Production silicon
Native Data Type	32-bit
Architecture Type	RISC
L1 Instruction Cache Size (kbyte)	16
L1 Data Cache Size (kbyte)	16
External Data Bus Width (bits)	32
Memory Clock (mhz)	133 MHz
Memory Configuration	4-1-1-1
L2 Cache Size (kbyte)	128 KB
L2 Cache Clock	266 MHz
Compiler Information	
Compiler Model and Version	Arm RealView Compilation Tools v3.1 build 569
Floating Point	Hardware

25

25

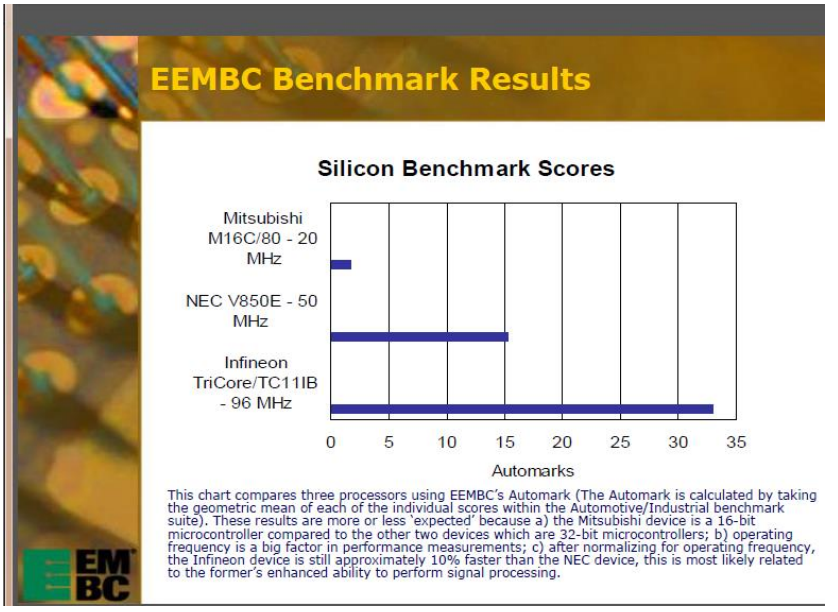
A report (II)
Freescale i.MX31-532MHz

Benchmark Scores	Iterations *per second*
Angle to Time Conversion	528925.03
Basic floating point	303782.18
Bit Manipulation	11307.02
Cache Buster	2671384.64
Response to Remote Request(CAN)	3098804.15
Fast Fourier Transform (Auto/Indust. Version)	614.51
Finite Impulse Response Filter (Auto/Indust. Vers)	190303.63
Inverse discrete cosine transform	10860.30
Infinite Impulse Response Filter	137289.71
Inverse Fast Fourier Transform (Auto/Indust. Vers)	659.47
Matrix arithmetic	1039.70
Pointer Chasing	11708.92
Pulse Width Modulation	2179341.27
Road Speed Calculation	2188980.08
Table Lookup and Interpolation	244956.19
Tooth To Spark	101902.74
Automark™	258.3

26

26

Benchmark results



Desktop Benchmarks

A processor throughput-oriented benchmark

Desktop Benchmarks

Standard Performance Evaluation Corporation

Processor-intensive benchmarks and graphics-intensive benchmarks

SPEC created a benchmark set focusing on processor: SPEC CPU2006.

- SpecInt2006, 12 integer benchmarks
- SpecFp2006, 17 floating-point benchmarks

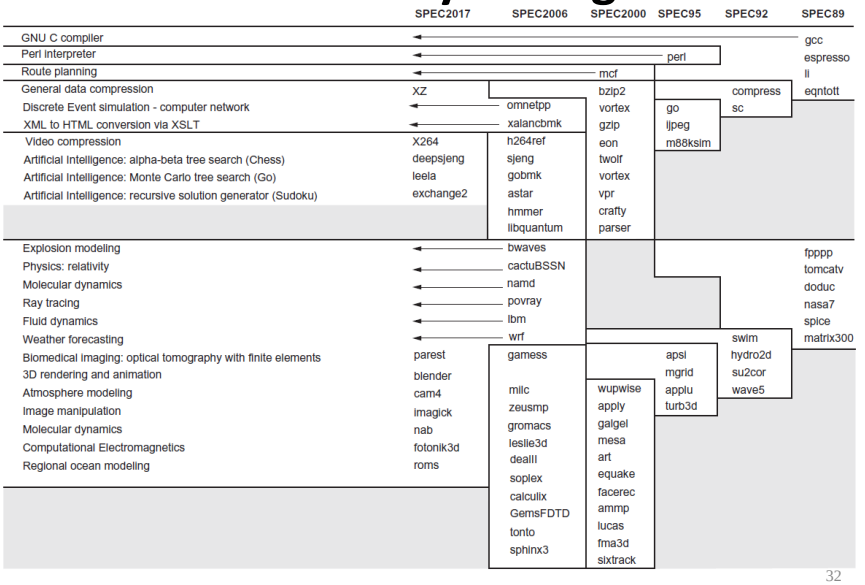
CPU: Integer Benchmarks

400.perlbench	C	PERL Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence: go
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence: chess
462.libquantum	C	Physics: Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

CPU: Floating Point Benchmarks

410.bwaves	Fortran	Fluid Dynamics
416.gamess	Fortran	Quantum Chemistry
433.milc	C	Physics: Quantum Chromodynamics
434.zeusmp	Fortran	Physics / CFD
435.gromacs	C/Fortran	Biochemistry/Molecular Dynamics
436.cactusADM	C/Fortran	Physics / General Relativity
437.leslie3d	Fortran	Fluid Dynamics
444.namd	C++	Biology / Molecular Dynamics
447.dealII	C++	Finite Element Analysis
450.soplex	C++	Linear Programming, Optimization
453.povray	C++	Image Ray-tracing
454.calculix	C/Fortran	Structural Mechanics
459.GemsFDTD	Fortran	Computational Electromagnetics
465.tonto	Fortran	Quantum Chemistry
470.lbm	C	Fluid Dynamics
481.wrf	C/Fortran	Weather Prediction
482.sphinx3	C	Speech recognition

Benchmark name by SPEC generation



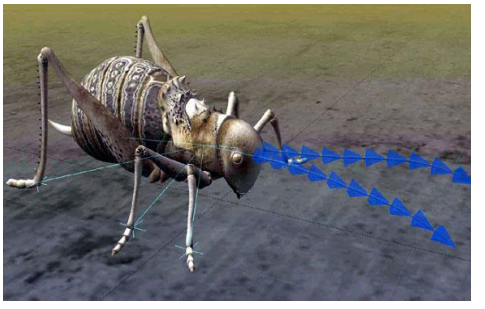
SPECspeed Metrics

- The elapsed time in seconds for each of the benchmarks is given and the ratio to the reference machine is calculated.
  - a Sun UltraSparc II system at 296MHz.
- The metrics are calculated as:
  - a Geometric Mean of the individual ratios,
  - each ratio is based on the median execution time from three runs.

33

Graphics and Workstation Performance

The SPECgpc^{SM} is a totally new graphics performance evaluation software.
Among the major changes are a new GUI, fully updated viewsets traced from newer versions of applications, larger models, and advanced OpenGL functionality such as shading and vertex buffer objects.



34

Precision Workstation R5500 (I)

Graphics Hardware Configuration	
Graphics Accelerator	NVIDIA Quadro 6000
Total Graphics Memory	6GB
Display Manufacturer/Model	E2211H
Display Resolution	1920 X 1080
Display Size/Technology	21.5" LCD
Display Refresh Rate	60 Hz
Swap on Vertical Retrace	OFF

35

Precision Workstation R5500 (II)

System Hardware Configuration	
Processor Type	Intel® Xeon® X5690
Processor Base Frequency	3.46GHz
Processor Characteristics	Turbo up to 3.73GHz
Number of Populated Processor Sockets	1
Cores per Processor Socket	6
Threads per core	1
Primary Cache per Core (KB)	32k (I) / 32k (D)
Secondary Cache per Socket (KB)	1536K
Tertiary Cache per Socket (KB)	12MB
System Memory (MB)	6144
Memory Type	DDR3 ECC
Memory Speed	1333
Memory Configuration	3 x 2048
Disk (GB)	500
Disk Interface	SATA
Disk RPM	7200

36

Precision Workstation R5500 (III)

Software Configuration	
Operating System	Microsoft Windows 7 Professional 64bit SP1
Compiler Name	Visual Studio 2008
Compiler Version	9.0.21022.8
Window System	MS Windows
OpenGL Version	4.0.0
OpenGL Renderer	Quadro 6000/PCI/SSE2
OpenGL Vendor	NVIDIA Corporation
Driver Version	259.57
Viewperf Version	11.0
Viewperf Executable	Standard
Price, availability, etc	
Price	$10,072
System Class	Single Supplier
Test Date	4/26/2011
General Availability	5/3/2011
Submitted by	Dell
Comments	
System BIOS is A00, HT disabled, turbo enabled, Aero disabled. System can be ordered through the Dell site(click here). Driver can be downloaded here (click here)	

Precision Workstation R5500 (IV)

Viewperf 11.0

Precision Workstation R5500 NVIDIA Quadro 6000		
Test #	**Weight (%)**	**Frames/sec**
1	12.00	49.20
2	12.00	21.80
3	14.00	19.80
4	14.00	33.20
5	12.00	106.00
6	12.00	83.40
7	12.00	71.30
8	12.00	123.00
Weighted Geometric Mean = 50.68		

Server Benchmarks

39

Standard Performance Evaluation Corporation (I)

Benchmarks

- CPU
- Graphics/Workstations
- MPI/OMP
- Java Client/Server
- Mail Servers
- Network File System
- Power
- SIP
- SOA
- Virtualization
- Web Servers

High Performance Computing, OpenMP, MPI

SPEC MPI2007
MPI2007 is SPEC's benchmark suite for evaluating MPI-parallel, floating point, compute intensive performance across a wide range of cluster and SMP hardware.

SPEC OMP2012
Designed for measuring performance using applications based on the OpenMP 3.1 standard for shared-memory parallel processing.

40

Standard Performance Evaluation Corporation (II)

Benchmarks

- CPU
- Graphics/Workstations
- MPI/OMP
- Java Client/Server
- Mail Servers
- Network File System
- Power
- SIP
- SOA
- Virtualization
- Web Servers

SOA

SPEC for typical middleware, database and hardware deployments of applications based on the Service Oriented Architecture.

Virtualization

SPEC's first benchmark addressing performance evaluation of datacenter servers used in virtualized server.

.

41

Server Benchmarks

Transaction Processing Council (TPC)

42

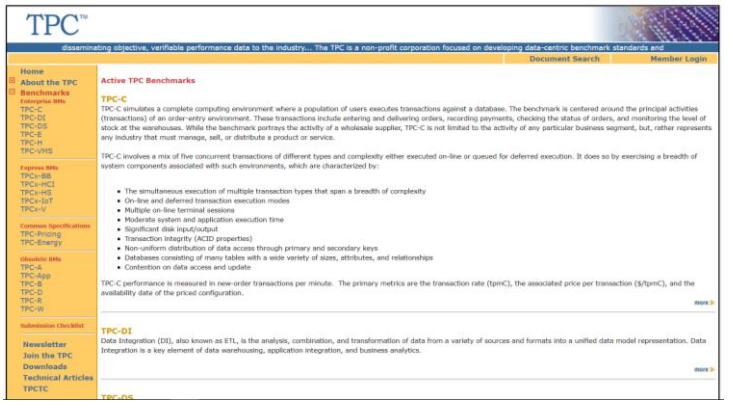
Transaction-processing (TP) benchmarks

Transaction-processing (TP) benchmarks measure the ability of a system to **handle transactions that consist of database accesses and updates**.

Airline reservation systems and bank ATM systems are typical simple examples of TP; more sophisticated TP systems involve complex databases and decision-making.

Transaction Processing Council (TPC)

TPC-C simulates **a complex query environment**.
TPC-H models **ad hoc decision support**.
TPC-E is a new **On-Line Transaction Processing (OLTP)** workload that simulates customer accounts.



TPC-C simulates a complex query environment

TPC-C simulates a complete computing environment where a population of **users** executes transactions against a **database**.

The benchmark is centered around the principal activities (transactions) of an order-entry environment.

These transactions include entering and delivering orders, recording payments, checking the status of orders, and monitoring the level of stock at the warehouses.

The benchmark portrays the activity of a wholesale supplier, represents any industry that must manage, sell, or distribute a product or service.

TPC-H is a Decision Support Benchmark

The TPC Benchmark™H (TPC-H) consists of a suite of business oriented ad-hoc queries and concurrent data modifications.

The queries and the data populating the database have been chosen to have broad industry-wide relevance.

This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions.

The performance metric reported by TPC-H is called the TPC-H Composite **Query-per-Hour Performance Metric** (QphH@Size).

TPC-E, On-Line Transaction Processing Benchmark

TPC-E involves a mix of twelve concurrent transactions of different types and complexity, either executed on-line or triggered by price or time criteria.

The database is comprised of thirty-three tables with a wide range of columns, cardinality, and scaling properties.

TPC-E is measured in **transactions per second** (tpsE).

TPC-E is not limited to the activity of any particular business segment, but rather represents any industry that must report upon and execute transactions of a financial nature.

PRINCIPLES OF COMPUTER DESIGN

Defining Computer Architecture

- The tasks of a computer designer are:
  - To list the most important features;
  - To design a computer **maximizing the performance in case of a set of applications** and energy efficiency respecting the **cost**, **power**, and **availability** constraints.

Computer Architecture

- **Instruction set,**
- **functional organization and logic design, and**
- **implementation.**
  - The implementation may encompass integrated circuit design, packaging, power, and cooling.

Optimizing the design requires familiarity with a very wide range of technologies, from compilers and operating systems to logic design and packaging.

Performance evaluation

- The Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

$$\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}$$

$$\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}} = \frac{\text{Seconds}}{\text{Program}} = \text{CPU time}$$

Principles of Computer Design

Different instruction types having different CPIs

Given a specific program instruction **i**

$$\text{CPU clock cycles} = \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i$$

CI*i* is the number of times for instruction i
CPI*i* is the number of clocks for instruction i

$$\text{CPU time} = \left(\sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time}$$

CPU time ~ N x CPI x Clock cycle time

CPU times for different implementations of the same architecture are proportional to the CPIs.

51

51

CPI: i7 vs ARM 53

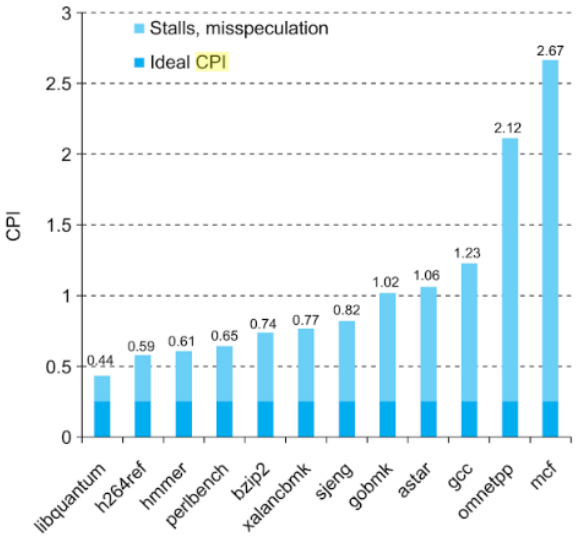
- The i7 uses an aggressive out-of-order speculative microarchitecture with deep pipelines with **the goal of achieving high instruction throughput by combining multiple issue and high clock rates**.
- The A53 uses a shallow pipeline and a reasonably aggressive branch predictor, leading to modest pipeline losses, while allowing the processor to **achieve high clock rates at modest power consumption**.
- In comparison with the i7, the A53 consumes approximately 1/200 the power for a quad core processor.

52

52

Intel i7 920 running SPEC2006 integer benchmarks

CPU time ~ N x CPI x Clock cycle time

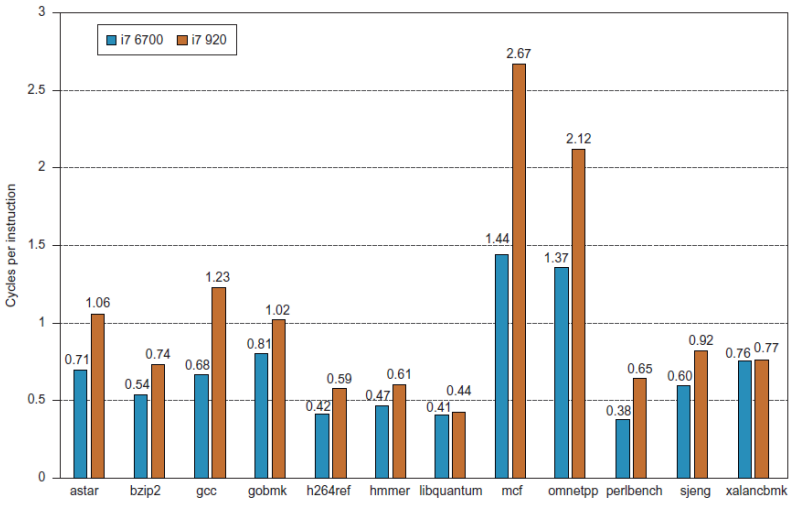


omnetpp,
Discrete Event Simulation

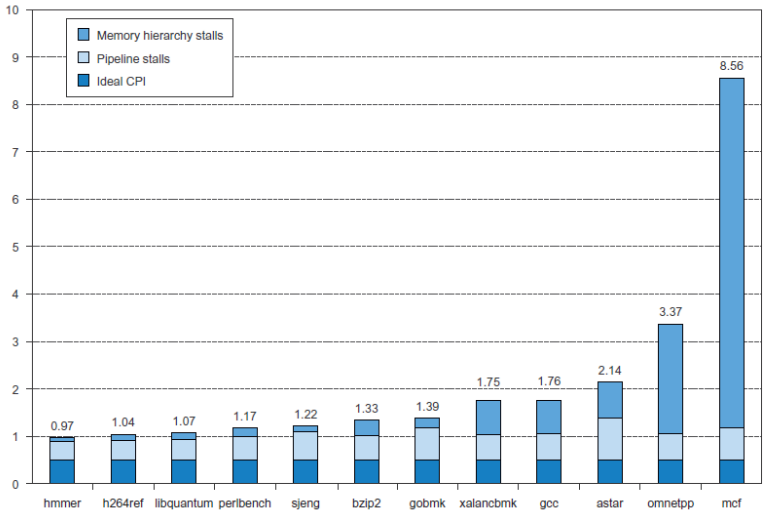
Uses the OMNet++ discrete event simulator to model a large Ethernet campus network.

mcf,
Combinatorial Optimization Vehicle scheduling.
Uses a network simplex algorithm (which is also used in commercial products) to schedule public transport.

The CPI for the SPECint2006 benchmarks on the i7 6700 and the i7 920



The estimated composition of the CPI on the ARM A53



Principles of Computer Design (I)

- Take Advantage of Parallelism
  - Use multiple processors, disks, memory banks, pipelining, multiple functional units
  - Design efficient parallel solution
- Principle of Locality
  - Reuse of data and instructions
- Focus on the Common Case
  - Amdahl's Law
- Take advantage of asynchronous service requests
  - Design to increase the percentage of asynchronous service requests
  - Cover the delay induced by synchronous operations with other activities

Principles of Computer Design (I)

- **Take Advantage of Parallelism**
  - **Use multiple processors, disks, memory banks, pipelining, multiple functional units**
  - **Design efficient parallel solutions**
    - **Ensure load balancing of each component (processor)**
    - **Minimize the overhead due to parallelism**
      - **time spent on communications (amount of data exchanged, speed of communication technology, private/shared link, traffic, …)**
      - **time spent to manage the parallelism (processes, distribution of work, collection of results, …)**

Principles of Computer Design (II)

- **Principle of Locality**
  - **Reuse of data and instructions:**
    - Temporal locality states that recently accessed items are likely to be accessed in the near future.
    - Spatial locality says that items whose addresses are near one another tend to be referenced close together in time.

Principles of Computer Design (III)

- Focus on the Common Case

  We have to decide:

  – 1. what the frequent case is and

  – 2. how much performance can be improved by making that case faster.

Amdahl's Law (I)

$$\text{Speedup overall} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

- **Fraction_e**, *The fraction of the computation time in the original solution that can be converted to take advantage of the enhancement*
- **Speedup_e**, The improvement gained by the enhanced execution mode for Fraction=e,

(Execution time for entire task using the enhancement when possible) **=**
(Execution time for **fraction of task** without using the enhancemen) +
(Execution time for **fraction of task** using the enhancement) =

$$(1\text{-Fraction\_e}) * T\_old + Fraction\_e * \frac{T\_old}{Speedup\_e}$$

$$\text{Speedup overall} = \frac{1}{(1\text{-Fraction\_e}) + \frac{Fraction\_e}{Speedup\_e}}$$

$$\text{Speedup overall} = \frac{1}{(1-\text{Fraction\_e}) + \frac{\text{Fraction\_e}}{\text{Speedup\_e}}}$$

Amdahl's Law (II)

Solution	Enhancement	Computation	Waiting for I/O	Overall time
Original	…	7 s	3 s	10 s
Disks	I/O Speedup = 3	7 s	1 s	8 s
CPU 1	CPU speedup =2	3,5 s	3 s	6,5 s
CPU 2	CPU speedup = 3	2,33 s	3 s	5,33 s

Disks Fraction_e=3/10, Speedup_e=3, Speedup overall=10/8=1,25

CPU 1 Fraction_e=7/10, Speedup_e=2, Speedup overall=20/13=10/6,5=1,54

CPU 2 Fraction_e=7/10, Speedup_e=3, Speedup overall=30/16=10/6,5=1,88

Disks 100 Fraction_e=3/10, **Speedup_e=100**, Speedup overall=10/8=1,42

Principles of Computer Design (IV)

Take advantage of asynchronous service requests

- Design to increase the percentage of asynchronous service requests
- Cover the delay induced by synchronous operations with other activities.

  - Write
  - Load instruction
  - speculative execution