

Embedded Systeme – Optimierung des Energieverbrauchs in ARM® Cortex™-M basierten Low-Power Applikationen mit dem µTrace

Jedermann kennt sie - Embedded Systeme, die im Automobil, im Telefon und in einer Vielzahl elektronischer Geräte, den Alltag und menschliches Verhalten beeinflussen und verändern. In vielen dieser Geräte steckt mittlerweile ein kleiner Computer. Neben Funktionalität und Rechenleistung ist ein niedriger Energieverbrauch verstärkt bei den mobilen und batteriebetriebenen Systemen ein wichtiges Kriterium, das als Stand-by-Zeit oder Betriebszeit häufig über den Kauf und damit über den Erfolg eines Produktes entscheidet. Aber auch weil Strom und Temperatur Halbleiterbauteile belasten und die Alterung beschleunigen, ist es besonders für Systeme in temperaturkritischen Umgebungen wichtig, Verlustleistung und damit verbundene Eigenerwärmung zu reduzieren. Entschleunigte Alterung verlängert die Lebensdauer und erhöht so die Qualität der Produkte.

Heutige Halbleiterprodukte haben bereits im Design intelligente Versorgungskonzepte vorgesehen, mit denen applikationsabhängig der Verbrauch gesteuert werden kann. So können Frequenzen variiert, Module bei Nichtbenutzung abgeschaltet oder über Power-Down-Modi nahezu der gesamte Chip in Ruhezustand versetzt werden. Es ist Aufgabe der Software, diese Möglichkeiten der Hardware optimal einzusetzen und dabei den Energieverbrauch zu minimieren.

Die elektrische Energie setzt sich als Produkt aus den drei Parametern Strom, Spannung und Zeit zusammen, die in derartigen Systemen einer starken Dynamik unterworfen sind. Im iterativen Prozess der konstruktiven Optimierung muss es mit der messtechnischen Erfassung möglich sein, Rückschlüsse auf die Ursachen der Dynamik im Verlauf von Strom und Spannung ziehen zu können.

Dies gelingt, wenn die Verbrauchswerte dem Ablauf Applikation zugeordnet werden können. Anhand dieser Korrelation kann der Energieverbrauch in seiner Effizienz beurteilt und daraus geeignete Korrekturmaßnahmen abgeleitet werden, die z.B. den Speicher-Einsatz und die Verwendung interner und externer Ressourcen optimieren und die verbrauchsintensive Bauteile austauschen.

Keine leichte Aufgabe, die aber von TRACE32-Debuggern problemlos gelöst werden kann, da sie über die notwendigen Schnittstellen zum Target und auch über ausreichende Kapazitäten zur Speicherung der Trace- und Verbrauchsdaten verfügen.

Energy Profiling beschreibt die Methode, die Strom- und Spannungswerte mit den Debuggern von Lauterbach erfasst und über das Graphical User Interface (GUI) PowerView analysiert und grafisch darstellt.

Energy Profiling mit dem µTrace

Der µTrace ist Lauterbachs neuestes Produkt, das Debug- und Trace-Funktionen in einem Gerät integriert. Optimiert für ARM® Cortex™-Prozessoren unterstützt er eine Vielzahl von Derivaten unterschiedlicher Hersteller. Bild 1 zeigt den Messaufbau für das Energy Profiling mit dem µTrace. Der Debugger ist hier über 2 Kabel mit dem Target und über ein USB 3.0 - Interface mit dem Host-Rechner verbunden.

Das 20 polige Debug-Kabel bedient 2 Schnittstellen am Prozessor. Über JTAG/cJTAG/SWD werden Debugkommandos abgesetzt und damit der Programmfluss gesteuert und bei unterbrochener Programmausführung auf Systemressourcen zugegriffen. Auf diesem Wege werden über diese Schnittstelle die prozessorinternen Trace-Module (4bit-ETM) konfiguriert, die während der Laufzeit die Zugriffe der CPU auf Befehle und Daten über bis zu 4 Signale des gleichen Kabels nach außen protokollieren.

Mit einer Analog-Probe von Lauterbach werden über das zweite Kabel die Messwerte von Strom und Spannung eingelesen. Hier stehen 4 Kanäle zur Spannungsmessung und 3 Kanäle zur Strommessung zur Verfügung. Die Aufnahme der Stromwerte benötigt einen niederohmigen Shunt-Widerstand in der

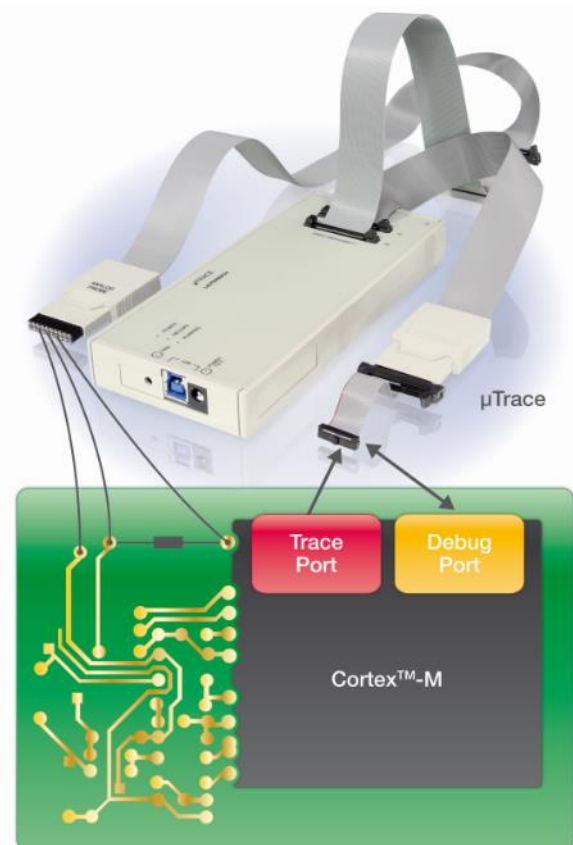


Bild 1: Energy Profiling mit dem µTrace

Versorgungsleitung auf dem Target, der nach Ohm'schen Gesetz einen dem Strom proportionalen Spannungsabfall verursacht. Die Probe kann jeweils 625000 Samples pro Sekunde erfassen, die anschließend mit einer Breite von 12bit digital dargestellt werden.

Dem μ Trace steht eine interne Kapazität von 256 MB zur Speicherung der Messreihen der Trace-Daten und der Strom- und Spannungswerte zur Verfügung. Die Reichweite des Speichers kann erweitert werden, in dem die Trace-Daten während der Laufzeit zum Hostrechner gestreamt werden.

Konfiguration des μ Trace zur Erfassung der Trace-Daten, der Strom- und Spannungswerte

PowerView ist die Benutzer-Oberfläche, die alle Funktionen zur Konfiguration des Debuggers und zur Anzeige der Ergebnisse bereitstellt. Die Settings können sowohl grafisch über die jeweiligen Dialoge als auch kommando-basiert vorgenommen werden. Bei komplexen und umfangreichen Einstellungen können alle Kommandoingaben in einem Script-File abgespeichert und so auf bequeme Weise wiederholt ausgeführt werden.

Das Energy-Profilung erfordert nur einige wenige Einstellungen. Dabei müssen Debug- und Trace-Interface von Prozessor und Debugger aufeinander eingestellt und die Analog-Probe konfiguriert werden. Voraussetzung für alles Weitere ist ein funktionierender Debug-Kanal. Hier müssen μ Trace und Prozessor über das gleiche Interface JTAG/cJTAG/SWD miteinander kommunizieren.

Im zweiten Schritt werden dann die Trace-Funktionen konfiguriert. Beim ARM[®] Cortex[™]-M sind es Embedded Trace Macrocell (ETM), Instrumentation Trace Macrocell (ITM) und Trace Port Unit Interface (TPUI), die Inhalt der Trace-Daten und Portbreite definieren. Die Konfiguration der Macros erlaubt, dass Programm- und Data-Trace inklusive Zeitstempel ausgegeben werden können. Bei einer begrenzten Traceportbreite, die bei dieser Prozessorarchitektur mit maximal 4bit gegeben ist, sollten Inhalt und Menge der Trace-Daten der verfügbaren Bandbreite angepasst sein. Für eine genaue Korrelation sollte der Programm-Trace möglichst ausführlich sein.

Die Datenerfassung mit der Analog-Probe wird gemäß Bild 2 über ein separates Menü konfiguriert, in dem die Strom- und Spannungskanäle aktiviert sowie die Werte der Shunt-Widerstände eingegeben werden. Weiterhin kann für jeden Kanal mit dedizierten Trigger-Settings die Ausnutzung des verfügbaren Speichers optimiert werden. Ebenso können an dieser Stelle drei Leistungskanäle aktiviert werden, deren Werte sich allerdings nicht durch Messung, sondern durch Berechnung aus den Werten der zugehörigen Strom-

und Spannungskanäle ergeben. In diese Rechnung geht zur Ermittlung des Stromes der Wert des Shunt-Widerstandes mit ein. Vereinfachend kann für einen Leistungskanal die Spannung als konstant angenommen werden, was die Anforderungen an das Messsystem vereinfacht, aber auch die Genauigkeit der Leistungsberechnung reduziert. Damit sind alle notwendigen Vorbereitungen getroffen. Über den Debugger kann mit dem Start der Ausführung der Applikation oder durch definierte Triggerbedingungen die Aufzeichnung von Trace und Messwerten gestartet werden.

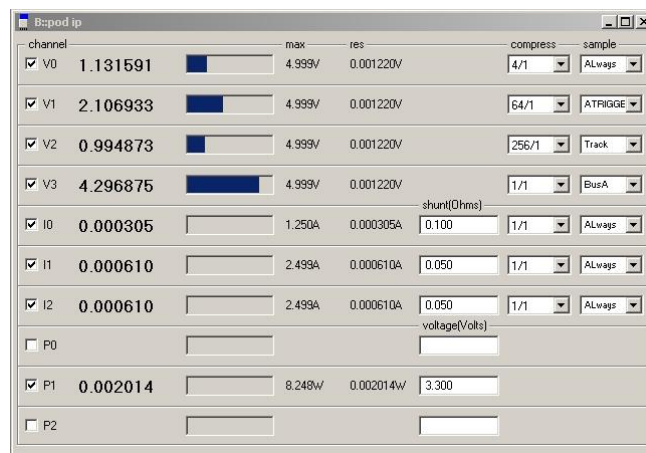


Bild 2: Konfiguration der Analog-Probe

PowerView – Korrelation der Messwerte und Auswertung der Ergebnisse

Während die Strom- und Spannungswerte kontinuierlich und äquidistant eingelesen werden, erfolgt die Übertragung der Trace-Daten komprimiert in Paketen versehen mit einem Zeitstempel vom Target. Nach Entpacken der Trace-Daten werden beide Messreihen mit Hilfe der μ Trace-internen Systemuhr auf eine gemeinsame Zeitachse umgerechnet und so jedem Verbrauchswert eine Position im zeitlichen Ablauf der Applikation zugeordnet. Diese Transformation der Zeitachsen beider Messreihen stellt als Bindeglied den Bezug zu den High-Level-Language (HLL) Source-Dateien her.

Die aufgezeichneten Trace-Daten und Messwerte können in unterschiedlicher Form als alphanumerische Auflistung und grafisch als Chart zeitlich korreliert in einzelnen Fenstern der GUI dargestellt werden. Bild 3 zeigt das Ergebnis einer Messung, bei der der durch Benutzung von externem Speicher verursachte Energieverbrauch untersucht wird. Während in diesem Beispiel eine Applikationschleife mehrmals nacheinander ausgeführt und dabei abwechselnd der Cache aktiviert und deaktiviert wird, werden der Versorgungsstrom von Memory-Interface und Core gemessen. Das obere Fenster von Bild 3 zeigt im

Diagramm die Laufzeit der verschachtelt aufgerufenen Funktionen. In dieser Anzeige wird die kürzere Laufzeit der schnelleren Zugriffe aufgrund des aktivierten Caches deutlich. Das Diagramm in der Mitte zeigt den Stromverlauf am Speicherinterface. Die höheren Stromwerte werden durch den Zugriff auf den externen Speicher im zweiten Durchlauf bei deaktiviertem Cache und vor dem dritten Durchlauf beim Nachladen des Caches verursacht. Die Verwendung des Caches wird auch am Versorgungsstrom des Cores sichtbar, dessen Zeitverlauf im unteren Fenster zu sehen ist. Die Zeitbereiche der einzelnen Fenster sind miteinander synchronisiert. Eine Änderung des Zeitbereichs des einen Fensters geschieht synchron auch bei den anderen Fenstern. So ist es auf einfache Art möglich, einzelne Strom- und Spannungswerte den entsprechenden Phasen der Applikation zuzuordnen. Aus den Messwerten können jedoch noch weitere Informationen berechnet oder statistisch abgeleitet und dargestellt werden. Dazu gehört die Berechnung der Verbrauchswerte der Leistungs Kanäle, die Ermittlung von Minimal- und Maximalwerten und funktionsbezogene Zuordnung des anteiligen Verbrauchs. Abschließend können alle aufgezeichneten Daten in Dateien gespeichert, später wieder geladen und mit dem Simulator auch offline analysiert werden.

Zeitstempel eine begrenzte Genauigkeit, die für eine Zuordnung auf Funktionsebene problemlos ausreicht, eine Zuordnung auf Befehlsebene aber nicht zulässt. Aufgrund der Befehlspipeline und Multicore-Architekturen werden Befehle generell parallel verarbeitet und ein Verbrauchswert kann nicht einem einzelnen Befehl zugeordnet werden. Weiterhin verursacht die Benutzung des Trace-Interfaces einen erhöhten Stromverbrauch, der durch das Mess-System ebenfalls mit erfasst wird und der bei hochgradigen Optimierungen durchaus den Messwert verfälschend ins Gewicht fallen kann.

Geachtet dessen ist das Energy-Profilierung mit dem μ Trace eine äußerst effektive Methode zur Verbrauchsoptimierung, die bei der Applikationsentwicklung, Compiler-Optimierung und der Chip-Inbetriebnahme sinnvoll eingesetzt werden kann.

Im Lieferumfang des μ Trace ist neben Netzteil das Debug- und USB-Kabel enthalten. Eine Reihe von Adaptern für das Debug-Kabel und die Analog-Probe sind als Zubehör erhältlich.

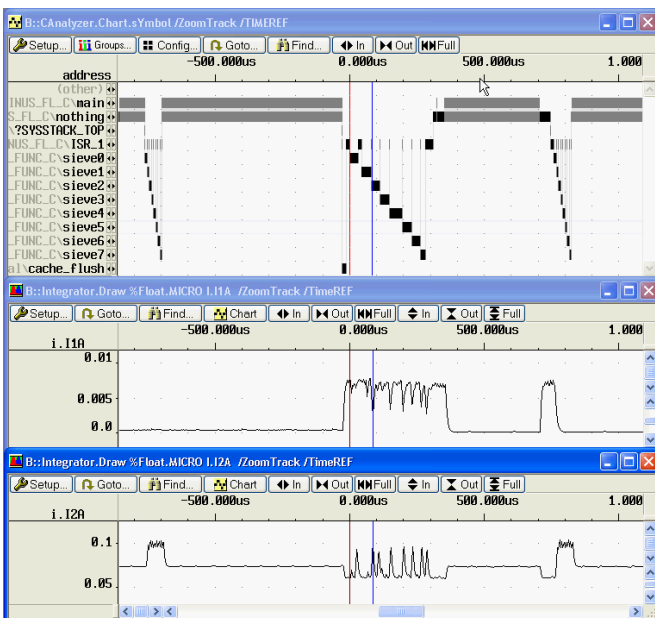


Bild 3: Darstellung der Ergebnisse mit PowerView

Mit dem μ Trace, der Analog-Probe und dem PowerView steht ein kompaktes System zur Verbrauchsbestimmung zur Verfügung, das von jedem Entwickler bequem gehandhabt und bedient werden kann. Allerdings hat diese Methode auch ihre Grenzen. Die Korrelation der Messwerte mit den Trace-Daten besitzt abhängig der vom Trace-Macro ausgegebenen

Lauterbach GmbH
 Altlaufstrasse 40
 85635 Hoehenkirchen, Germany
 Phone: +49 8102 9876-0
 Fax +49 8102 9876-187
 info@lauterbach.com

September 2013