eXtended eXternal Benchmarking eXtension (XXBX)

John Pham and Jens-Peter Kaps

Cryptographic Engineering Research Group (CERG)
http://cryptography.gmu.edu
Department of ECE, Volgenau School of Engineering,
George Mason University, Fairfax, VA, USA

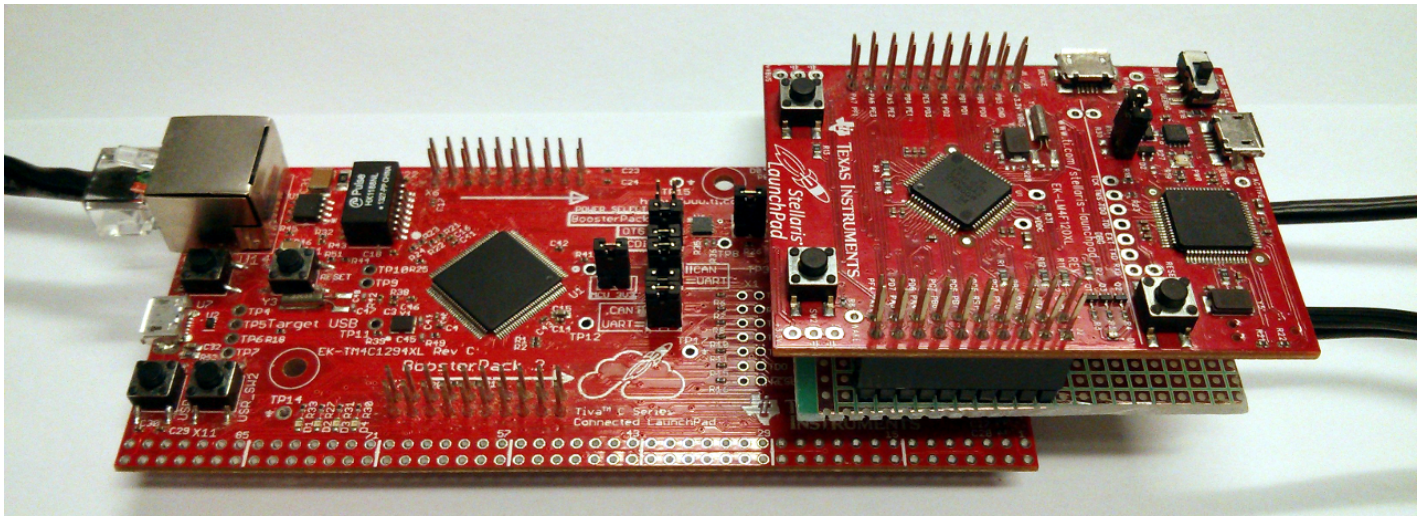
DIAC 2015

Outline

Introduction & Motivation

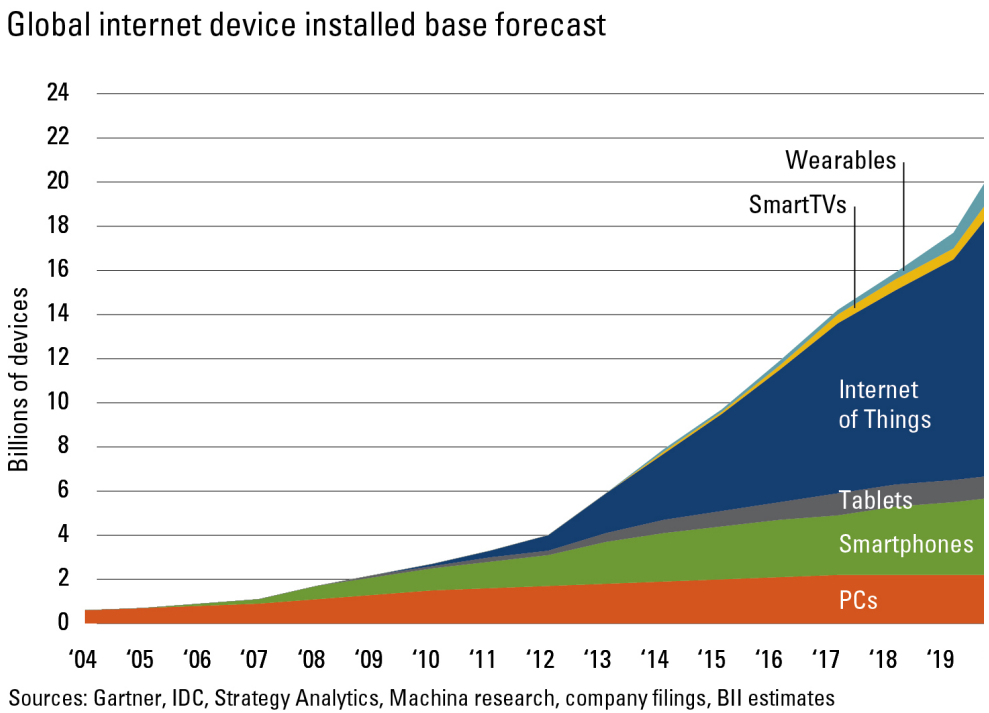
Introduction

- XXBX is a tool for benchmarking algorithms on microcontrollers that cannot efficiently run their own operating system and compilers.
- It uses the following Metrics:
  - Throughput - cycles per byte
  - ROM usage - bytes
  - RAM usage - bytes
  - Power - milliwatts



Motivation

- IoT promises a dramatic increase in devices, many will be microcontrollers or SOCs.
- 32-bit microcontrollers are projected to take lead over 8/16-bit by 2018.
- 51% of all 32-bit microcontrollers were ARM based in 2012.



Previous Work

SUPERCOP

- System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives.
- Benchmarks many implementations of many primitives across multiple operations on multiple hardware platforms.
- Supports environments capable of running Linux and hosting a compiler.
- Series of shell scripts and C test harnesses, and comprehensive collection of algorithm primitive implementations.
- Verifies correct execution of implementations and times cycles required per byte processed.
- Does not measure ROM and RAM usage or power consumption.

`http://bench.cr.yp.to/supercop.html`

XBX

- eXternal Benchmarking eXtension -extends SUPERCOP
- Automated testing on real microcontrollers
- Compatibility with SUPERCOP algorithm collection ("algopacks") and output format
- Low cost hardware and software
- Our contribution to original XBX was to port it to the MSP430 platform and provide results for SHA-3 finalists.
- Measures ROM and RAM usage. Does not measure power consumption.

XBX Components

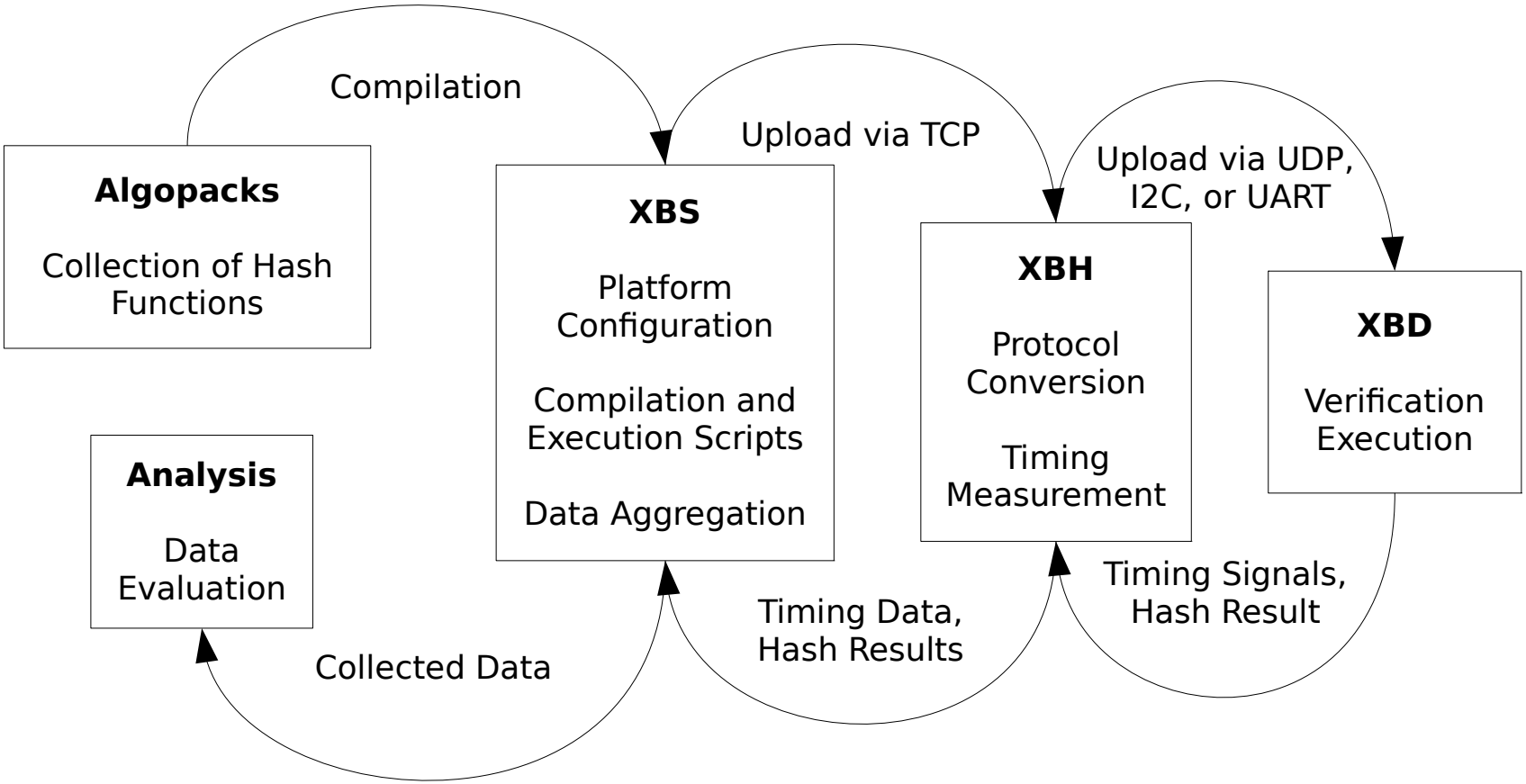


Figure: Block Diagram of XBX components

XBX Limitations

- Only supports hash functions
- No power measurements
- Does not use cycle counters
- Benchmarking takes a long time because embedded platforms are slow.
  - Simulation can run faster

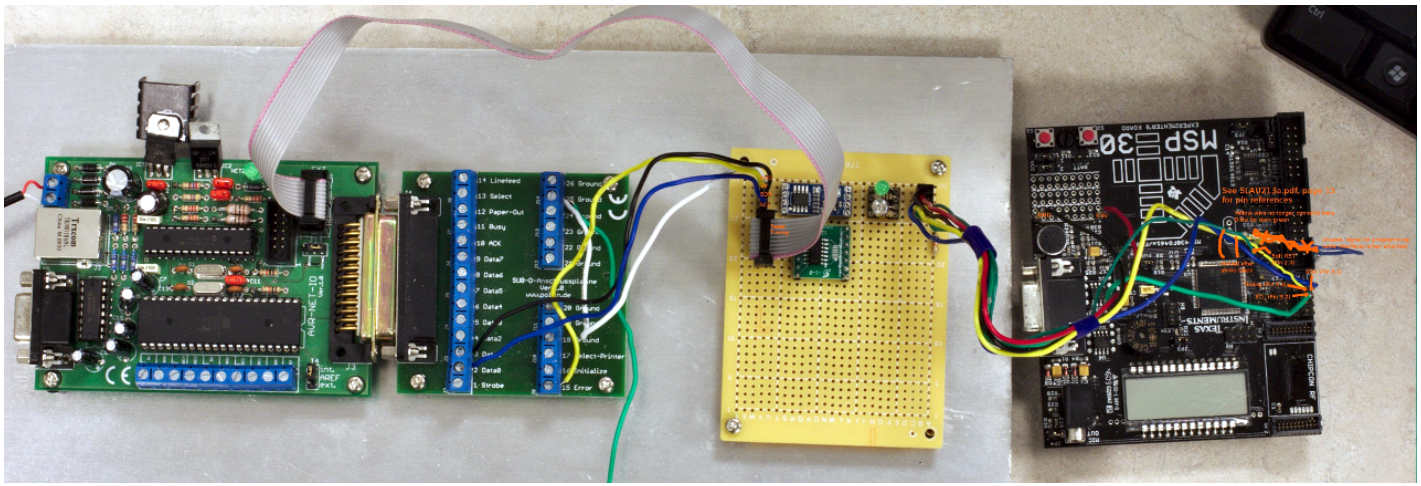


Figure: AVR-NET-IO ATmega32 board with MSP430

FELICS

- Fair Evaluation of Lightweight Cryptographic System
- Targeted for lightweight block ciphers
- Uses simulation when available else real hardware
- Supports Atmel AVR, MSP 430, ARM Cortex-M3
- Measures RAM, ROM, execution time.

`https://www.cryptolux.org/index.php/FELICS`

XXBX

Design Goals

- Add AEAD support
- Add power measurement
- Replace XBH in order to facilitate power measurement
- Add resuming partial runs
- Avoid breaking when Link-Time Optimization is enabled
- XBXX? :)

XBH Replacements

- Requires ethernet and I/O to XBD
- Hardware under initial consideration
  - Raspberry Pi
  - Beaglebone
- Linux-based boards very fast, but do not easily meet real-time requirements
- Tiva C Connected Launchpad chosen when it became available
  - ARM Cortex-M4F with ethernet connectivity.
  - 256kiB of SRAM and 1MiB of ROM
  - Dual 12-bit ADCs capable of 2 MSPS
  - Easily worked on bare metal without an OS

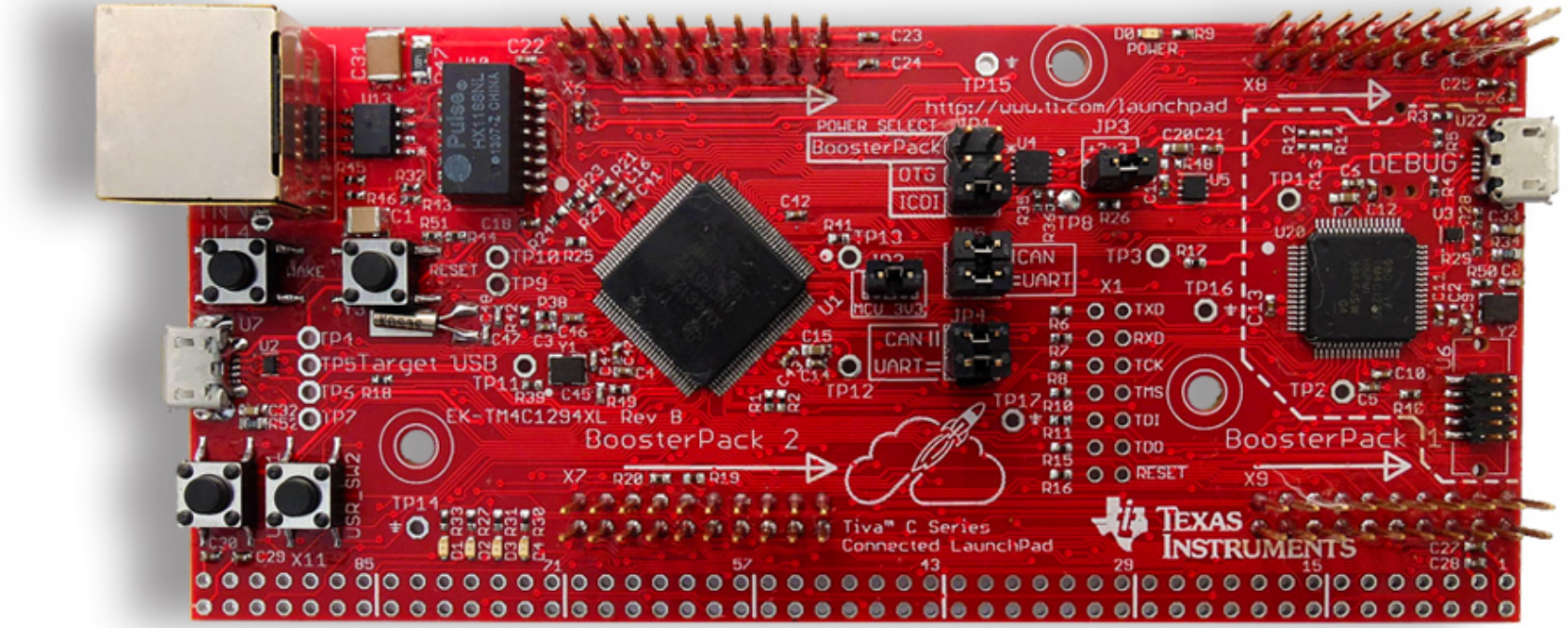


Figure: Tiva C Connected Launchpad (EK-TM4C1294XL)

XBD Hardware

- MSP-EXP430F5529LP
  - 16-bit MSP430 clockable to 25MHz, 10kiB SRAM and 128kiB flash
- EK-TM4C123GXL
  - 32-bit ARM Cortex M4F clockable to 80MHz, 32kiB SRAM and 128kiB flash
- Both of these newer versions of what is currently supported by XBX
- TODO: Support AVR, MIPS
- XBX also supports ARM Cortex-A which we don't intend to support yet. Regular SUPERCOP may run on some of these.

XBH Software

- Use FreeRTOS with LwIP instead of bare-metal
  - Easier multitasking- OS handles task switching instead of doing it explicitly
  - Easier to write network code - LwIP socket API can be used
  - LwIP and FreeRTOS port included in examples provided by Texas Instruments
- Original XBX used webserver-uvm from Ulrich Radig
- Hardware abstracted away

XBH code differences

- Only support TCP/IP for XBS $\leftrightarrow$ XBH communication. XBX supports both TCP/IP and UDP/IP.
- Only support I^2C for XBH $\leftrightarrow$ XBD
- Add length prefix to delimit messages
- Power measurements streamed to XBS

XBH code tasks

- lwIP TCP/IP
- XBH Server
- XBH command execution and XBD communication (same priority as XBH server)
- Ethernet Receive/Transmit - sends transmit and receive descriptors to lwIP
- Power Measurement - woken up periodically by timer interrupt to perform measurements and enqueuing them to the XBH server task.

XBH Interrupts

0. Unused
1. Timer Wraparound
2. Timer Capture
3. Max FreeRTOS SysCall Priority
3. Power Sample Timer
4. Watchdog
5. Unused
6. Unused
7. FreeRTOS kernel

XBD Software

- Largely the same as original XBX
- Replaced self-test implementation with SUPERCOP's
- Refactor out hash-specific code to make it easier to add other operations
- Add AEAD payload processing
  - XBH doesn't know anything about the operation under test, just routes it blindly to XBD from XBS.
  - XBD must know what is being in run order to unpack parameters and messages

XBH Software

- Completely rewritten in Python 3
- Now supports resuming runs if run fails and XBS crashes due to hung hardware
- Results now stored in a SQLite database
- Dropped unused features such as KAT-file verification and loading XBD in formats other than IHEX
- Builds performed in parallel

Current Sensing

- Measured by sensing voltage drop across a small shunt resistor
- High side
  - Directly measures current delivered by voltages source
  - Multiple ground paths do not need to be accounted for
  - No issues w/ ground loops
  - Must handle common-mode voltage
- Low side
  - Can be single-ended
  - Does not have to deal with common mode voltage
- We chose the high side configuration, as I/O pins could provide alternate ground paths causing measurement errors.

High side vs Low side

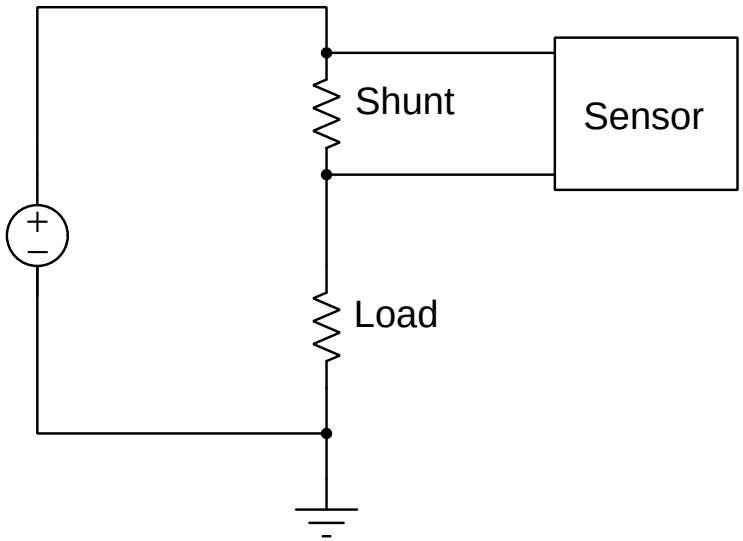


Figure: High side

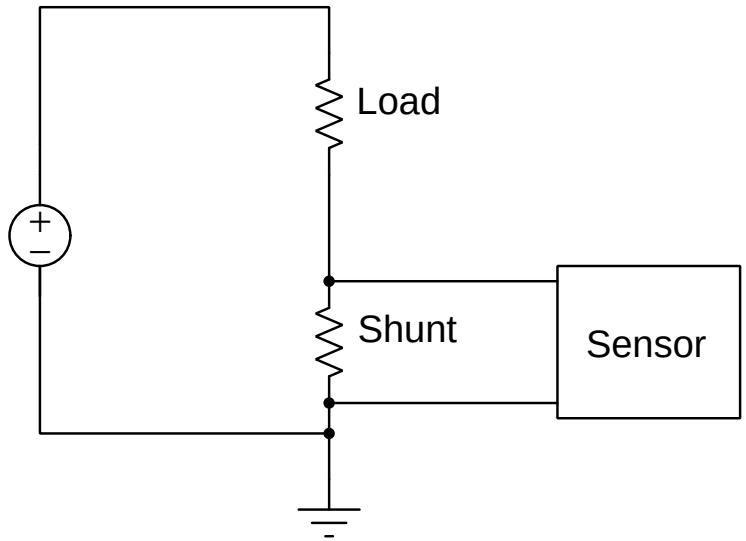


Figure: Low side

Current Sensor

- Utilize ADCs on Launchpad
  - These ADCs have input low-impedance, must be buffered
  - Need amplification, as shunt drop is low
- Considered putting op-amp in front of ADCs
  - Requires precision resistor network
  - More parts to deal with
- Use current sense amplifier in front of ADC - specifically INA225
  - Selectable gain to adjust for different target devices in different ranges (25-200), buffered output to deal with low ADC input impedance

INA225

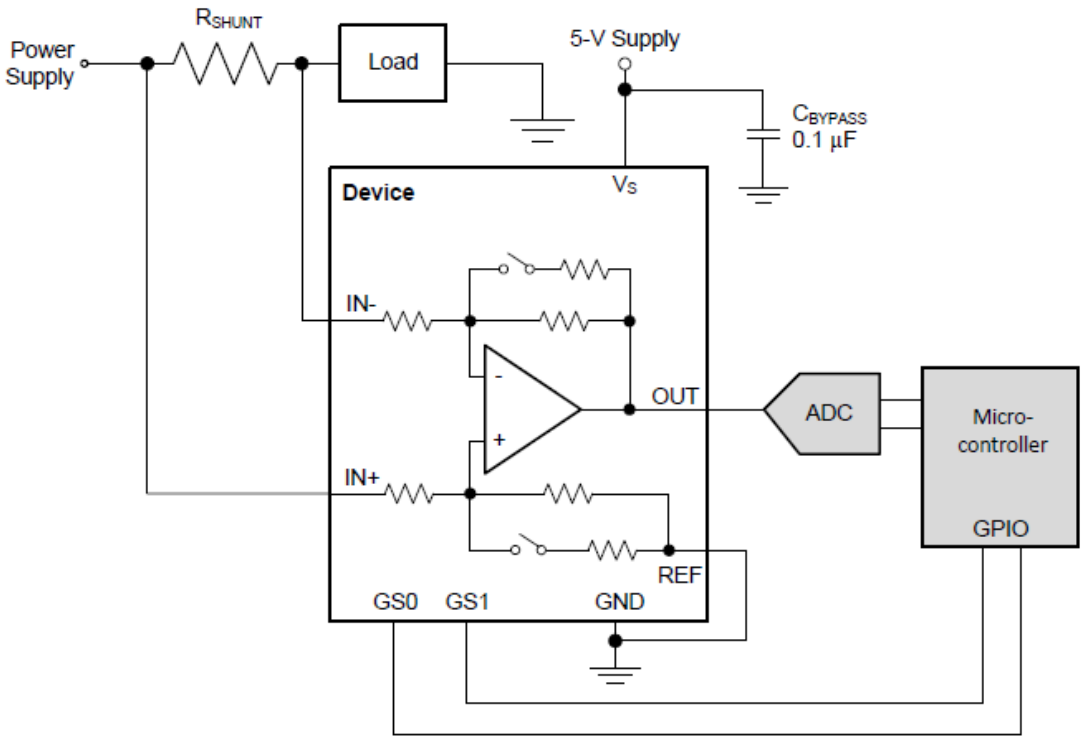


Figure: Power measurement circuit using INA225

Conclusions and Future Work

Conclusions

- XBX extended to include support for AEAD
- Enables benchmarking of power
- Allows resuming partial runs

SUPERCOP, XBX, XXBX Feature Comparison

	SUPERCOP	XBX	XXBX
Target Platform	Desktop/Server	Embedded	Embedded
Speed Benchmarks	✓	✓	✓
Memory Benchmarks		✓	✓
ROM Benchmarks	N/A	✓	✓
Supports AEAD	✓		✓
Power Benchmarks			✓

XBX-XBD and XXBX-XBD Comparison

XBX Supports

Device	Chip		ISA		Bus	f	OS	Price
	Atmel	ATmega1284P	AVR		8-bit	20 MHz	bare	
Exp.Board	TI	MSP430FG4618	MSP430		16-bit	8 MHz	bare	$117
FritzBox	TI	AR7	MIPS32	4KEc	32-bit		Linux	$300
Artila M501	Atmel	AT91RM9200	ARM920T	ARMv4T	32-bit	180 MHz	Linux	$116
NSLU2	Intel	IXP420	XScale	ARMv5TE	32-bit	266 MHz	Linux	$90
	IXP	LPC1114	ARM Cortex-M0	ARMv6-M	32-bit	50 MHz	bare	
	TI	LM3S811	ARM Cortex-M3	ARMv7-M	32-bit	120 MHz	bare	
BeagleBoard	TI	DM3730	ARM Cortex-A8	ARMv7-A	32-bit	1 GHz	Linux	$89

XXBX Supports

Device	Chip		ISA		Bus	f	OS	Price
Launchpad	TI	MSP430FR6989	MSP430		16-bit	16 MHz	bare	$18
Launchpad	TI	TM4C123GXL	ARM Cortex-M4	ARMv7E-M	32-bit	80 MHz	bare	$13
Future	Atmel	ATmega1284P	AVR		8-bit	20 MHz	bare	
MikroE	Microchip	PIC32MX360F064H	MIPS32	M4K	32-bit	80 MHz	bare	$25

Remaining work

- Integrate the power measurement hardware
- Perform a full benchmarking run on all AEAD and hash algorithms that have implementations that can run
- Extend platform support to AVR and MIPS
- Documentation.

Thanks for your attention.