Asynchronous Charge Sharing Power Consistent Montgomery Multiplier

Jiaoyan Chen[1], Arnaud Tisserand[2], Emanuel Popovici[3], & Sorin Cotofana[1]

[1]Delft University of Technology, The Netherlands
[2]University Rennes 1, Lannion, France
[3]University College Cork, Ireland

21st IEEE International Symposium on Asynchronous Circuits and Systems
Mountain View, Silicon Valley
California, USA, May 4-6, 2015

Overview

- Introduction/Motivation
- Latch-less Asynchronous Charge Sharing Logic (LACSL)
- LACSL Montgomery Multiplier Implementation
- Experimental Results
- Conclusions

Introduction

Side Channel Attacks - Exploit Correlation

- Processed Internal Data
- Measured Parameter(s)

Why power Consistency is needed?

- Against Power Attacks
- Widely utilised: Differential Power Analysis (DPA)
  - Low Cost & Versatile

Why Montgomery Multiplier (MM)?

- (One of the) Most Popular Modular Multiplication for Elliptic Curve Cryptography (ECC), RSA.

Attack on Cryptographic System

Why Power is not Consistent?

- Highly data dependent, glitches, hazards, etc.
- Vulnerable to Diffirential Power Analysis.

Existing techniques, approaches?

- Data independent power consumption CMOS logic (no glitch occurrence)
- Only small circuits implemented, i.e., s-box, XOR gates, …

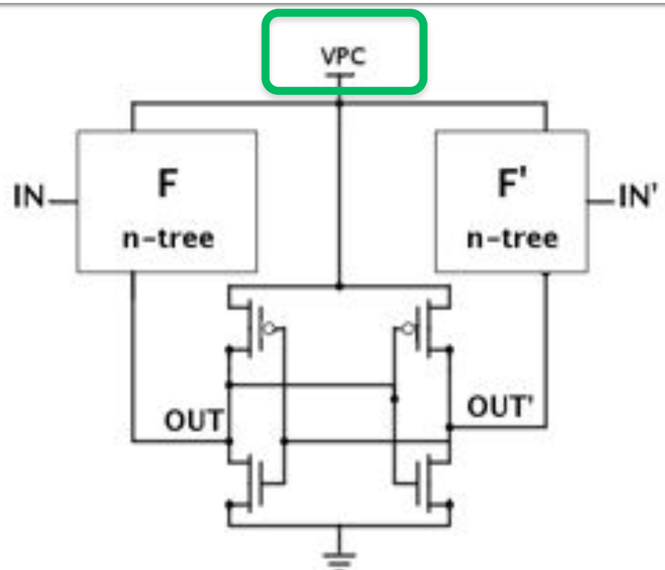
Latch-less Asynchronous Charge Sharing Logic (LACSL)

- Predecessor -- ACSL:

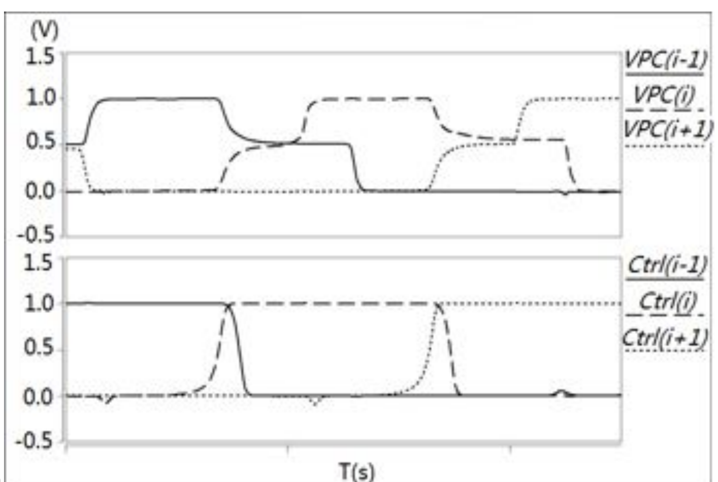
  Charge sharing: Low (dynamic, static) power.
  Dual-rail: robust against variations.

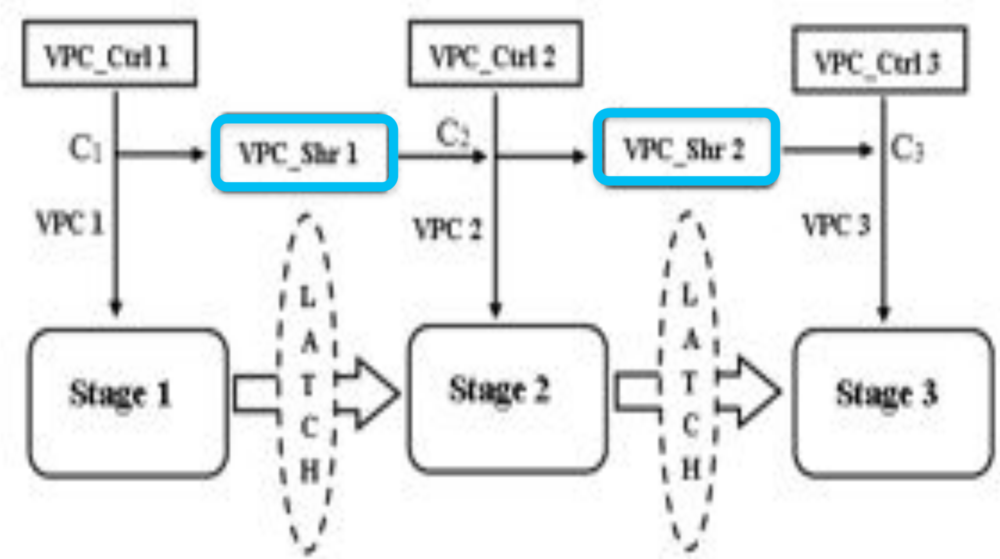
  But, latch still involved, data dependant.
  Also, power can still be reduced.

ACSL generic structure, block diagram and waveforms



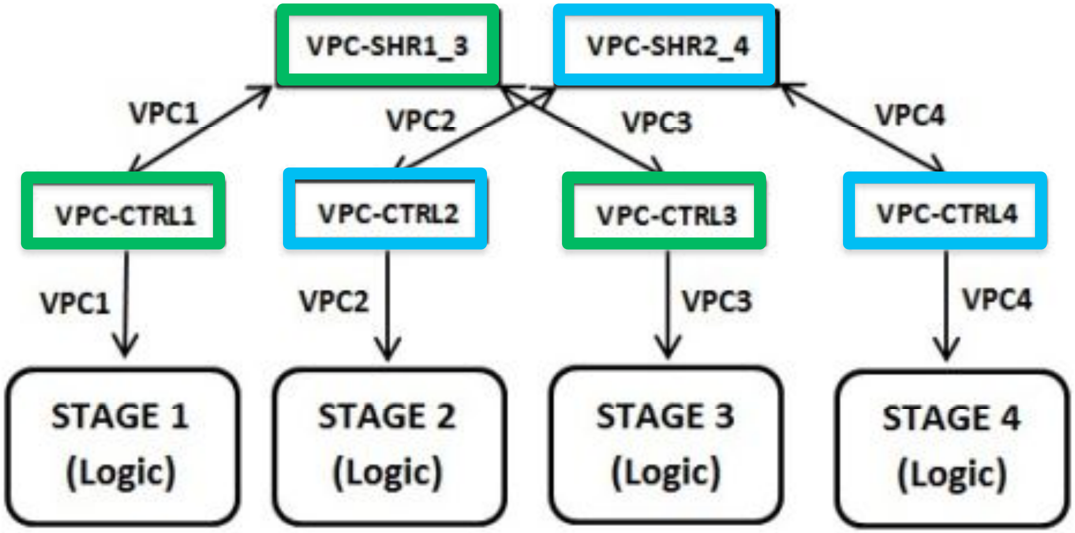
1. Voltage Power-Clock (VPC) controls the operation of ACSL. Inherited power gating.
2. Share voltage between neighboring stages.
3. Charging ➔Sharing ➔Discharging

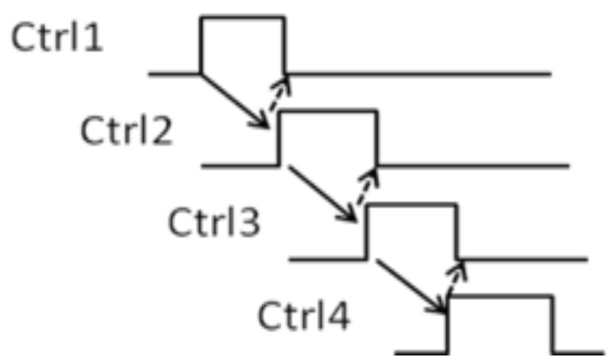




LACSL– How to spare latches?

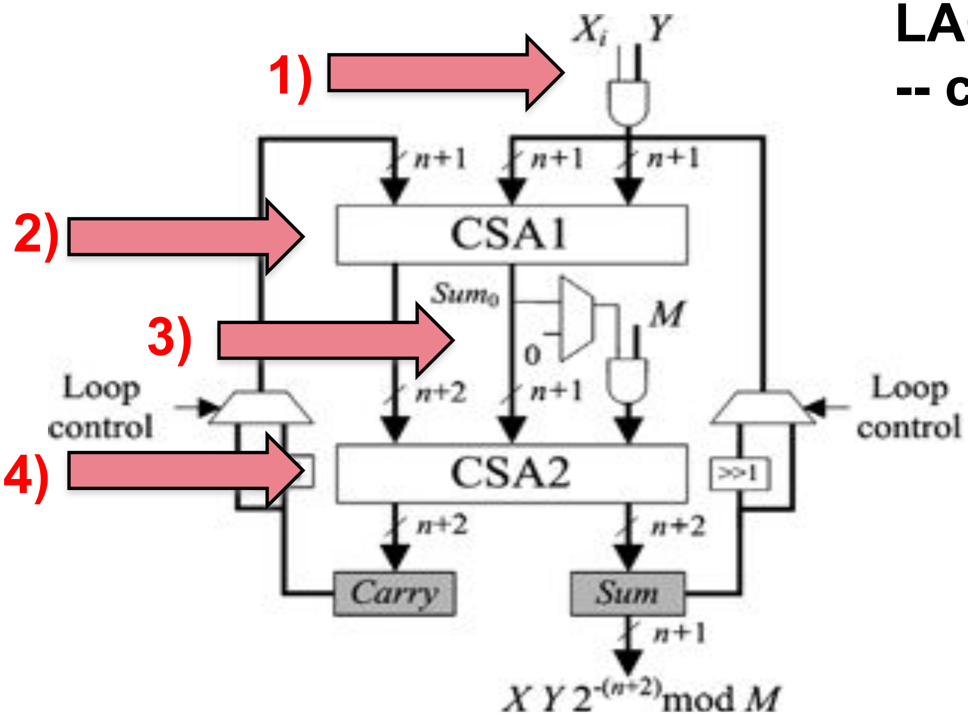
- We use interleaved charge sharing (at least one isolation stage). – No dedicated storage elements required yet signal validity preserved.
- However, the handshaking transition diagram remains the same as for ACSL. – no conflicts





LACSL Montgomery Multiplier

- Carry Save Adder Array Based



LACSL prefers well-balanced structure -- charge sharing effieciency

How many layers we have now?

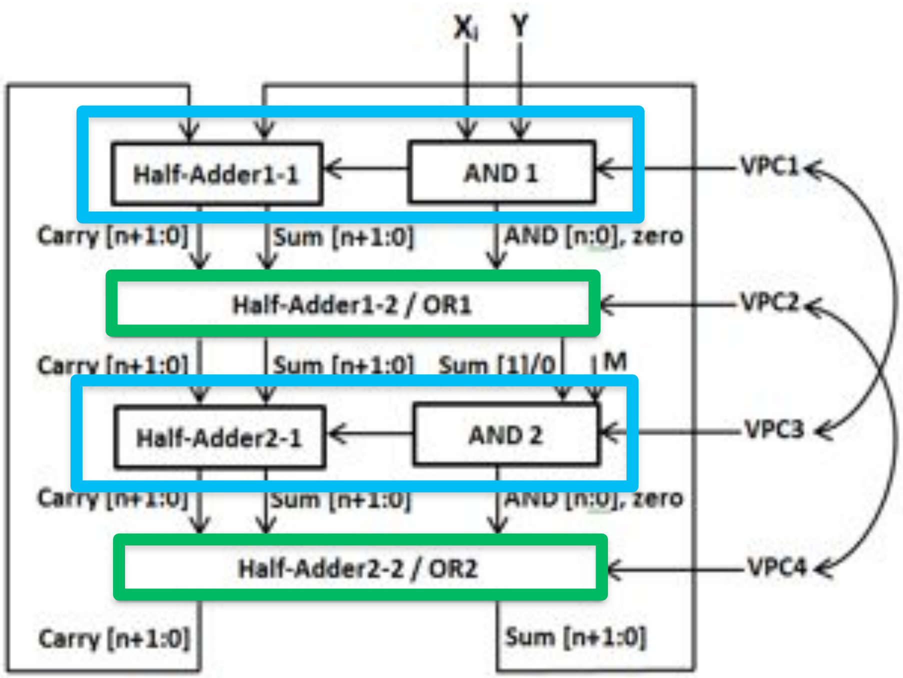
1) **AND**
2) **CSA1**
3) **AND**
4) **CSA2**
5) **Shifter – No need in LACSL, realized by simply wiring.**

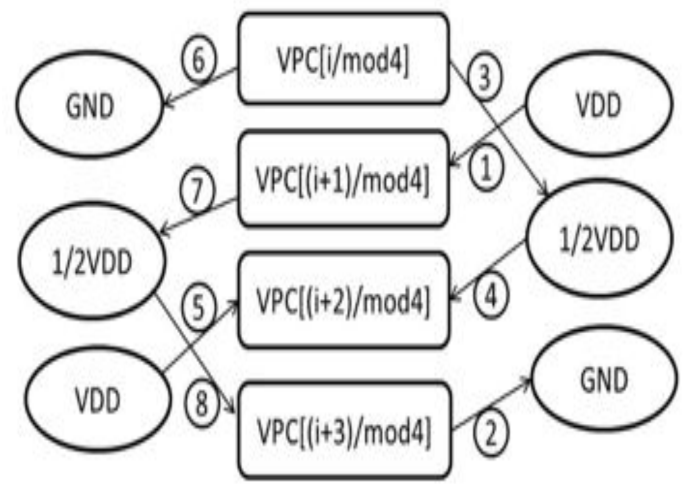
Unbalanced

LACSL Montgomery Multiplier Reform

- **How we change the formation?**

  Split CSA and mix with AND gates into a layer of Half adder HA & AND and a layer of HA & OR.

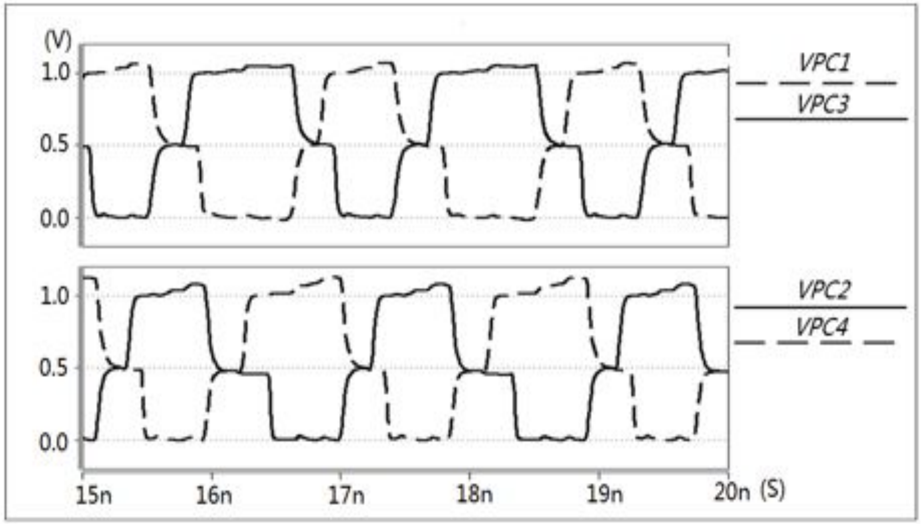




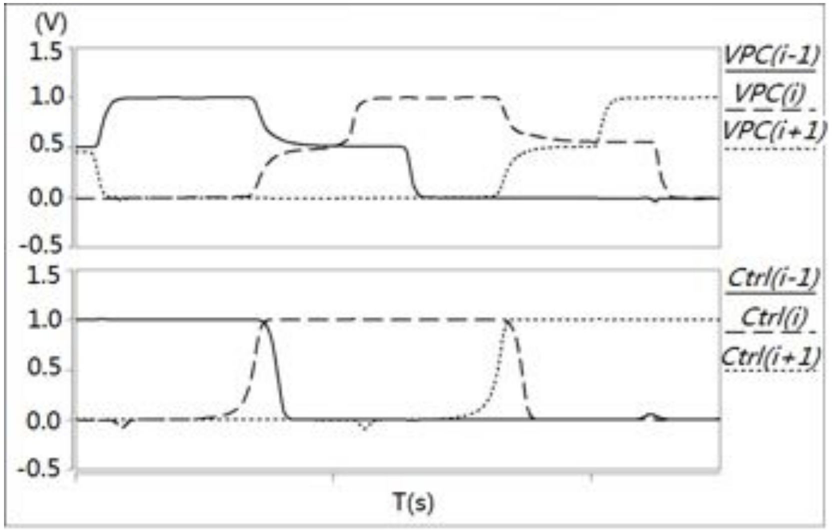
LACSL VPCs Operation FLow

LACSL Montgomery Multiplier Reform

- LACSL vs ACSL VPC waveforms



LACSL VPCs -- Leapfrog



ACSL VPCs -- Cascaded

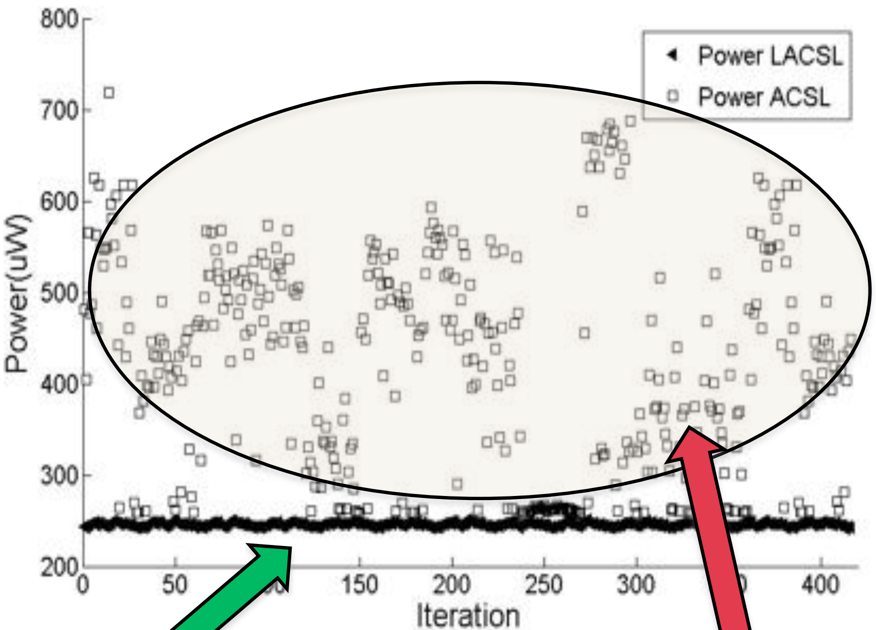
LACSL MM Implementation Results

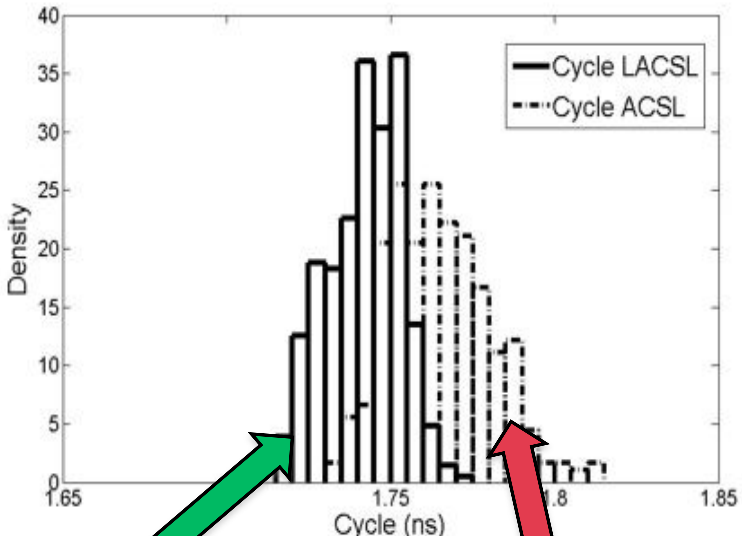
- 32-bit extensively invesitaged using HSPICE. Other bigger size, up to 256-bit also simulated. 45nm, VDD=1V

- For the 32-bit MM:
  1) fixed *X*, fixed *M*, various *Y* with different Hamming weights ranging from 10 to 22 -- Energy/Iteration
  2) 100 sets of random *X*, fixed *M*, random *Y* – *Energy/Operand*

- For the 64-bit, 128-bit, and 256-bit LACSL MMs, 10 iterations of random input vectors with corresponding bit-width are generated and simulated.

32-bit LACSL MM vs ACSL MM Power, Delay per Cycle





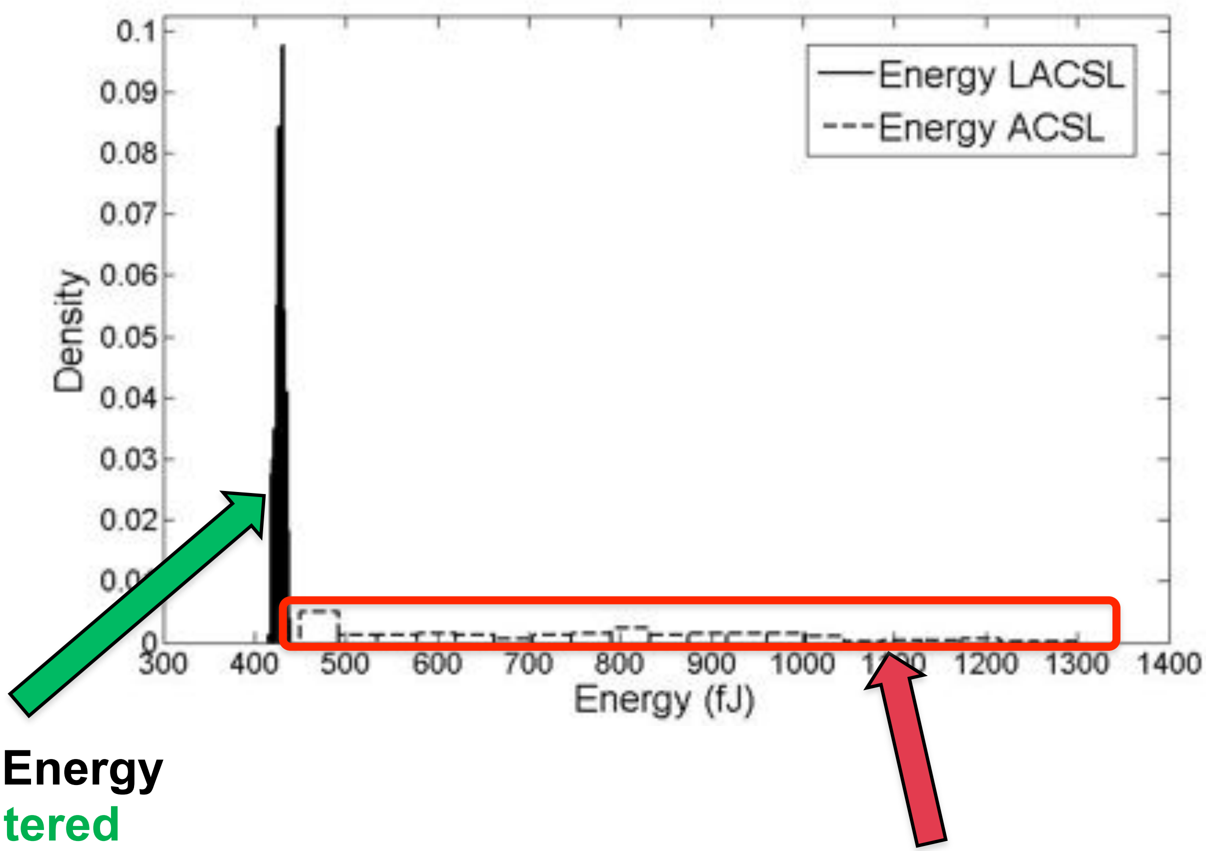
LACSL Power – Constant

ACSL Power – Scattered

LACSL Delay

ACSL Delay

32-bit LACSL MM vs ACSL MM Energy Cycle



LACSL Energy
– Centered

ACSL Energy
– Distributed

32-bit MMs Data Comparison

Comparison Metrics

- Maximum Energy Consumption [fJ]
- Minimum Energy Consumption [fJ]

- Normalized Energy Deviation (NED)

$$NED = \frac{Max(energy / cycle) - Min(energy / cycle)}{Max(energy / cycle)}$$

- Standard Deviation (SD) [fJ]
- Average Energy Consumption [fJ]
- Normalized Standard Deviation
- Leakage Power [μW]

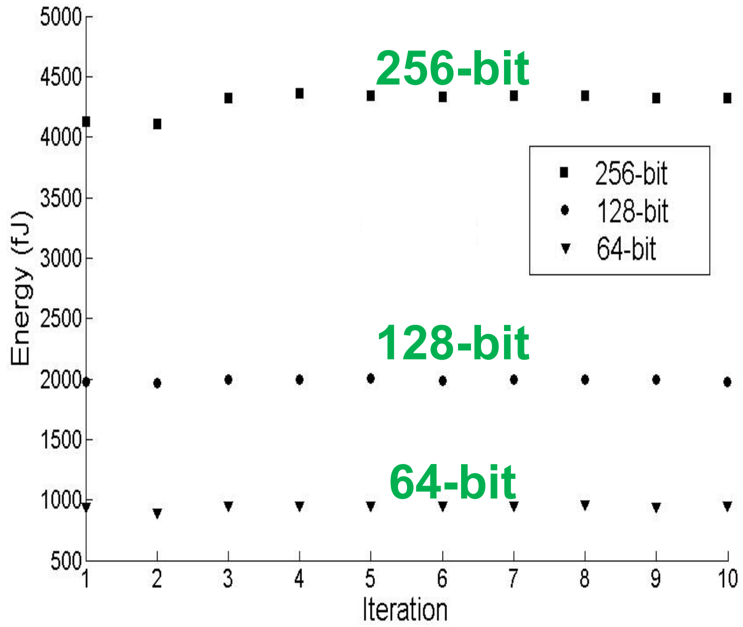
32-bit MMs Data Comparison (2)

32-bit MMs	ACSL per cycle	LACSL per cycle	LACSL per Operand X,Y
MAX (fJ)	1288	439	14832
MIN (fJ)	450	418	14698
NED	0.65	0.048	0.009
SD (fJ)	215	4.7	23
Mean (fJ)	733	428	14752
NSD	0.29	0.011	0.0015
Leakage Power (µW)	7.1	2.05	2.05

- **Less than 1% Normalized Energy Deviation**
- **~40% reduction of average energy consumption**
- **3.5 x leakage power reduction**

64, 128, 256-bit LACSL MMs Data -- Scalability

LACSL MMs	*64-bit*	*128-bit*	*256-bit*
Avg. Power (uW)	500	1046	1970
Avg. Cycle (ns)	1.9	2.0	2.1
Avg. Energy (fJ)	950	2092	4137
Avg. Energy Deviation	0.5%	0.5%	0.3%



The highest deviation is only 0.5% !

Conclusions

- Latch-less Asynchronous Charge Sharing Logic is based on ACSL without using the dedicated latches and thus it can achieve power consistency.
- Interleaving charge sharing is utilized to preserve data integrity.
- A LACSL Montgomery Multiplier is developed by spliting and mixing different layers of the original structure.
- Various LACSL MMs are simulated. High power/energy consistency is demonstrated.
- Normalised Energy Deviation less than 1%.
- 45% energy savings over ACSL MMs.
- 3.5x less leakage power over ACSL MMs.
- Good scalability is demonstrated.

Acknowledgements

- This work has been sponsored by the European Commission FP7 FET-Open iRISC (Innovative Reliable Chip Designs from Unreliable Components) project, SpiNaCH (CNRS PICS 6023) project, and PAVOIS project (ANR 12 BS02 002 01).
- Many thanks to the ASYNC 2014 anonymous reviewers!

Questions?

Thank you for your attention !