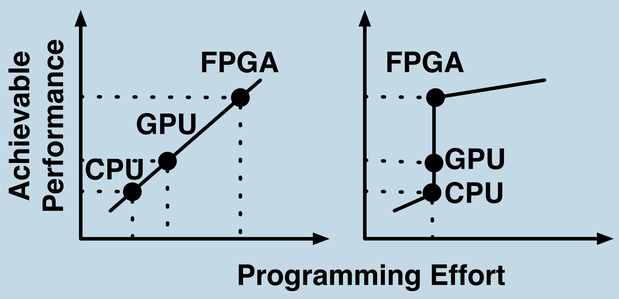
Towards High-Throughput with Low-Programming Effort: From General-Purpose Multicores to Dedicated Circuits

J. Andrade[1], G. Falcao[1], V. Silva[1], C. Antonopoulos[2], N. Bellas[2] and P. Ienne[3]

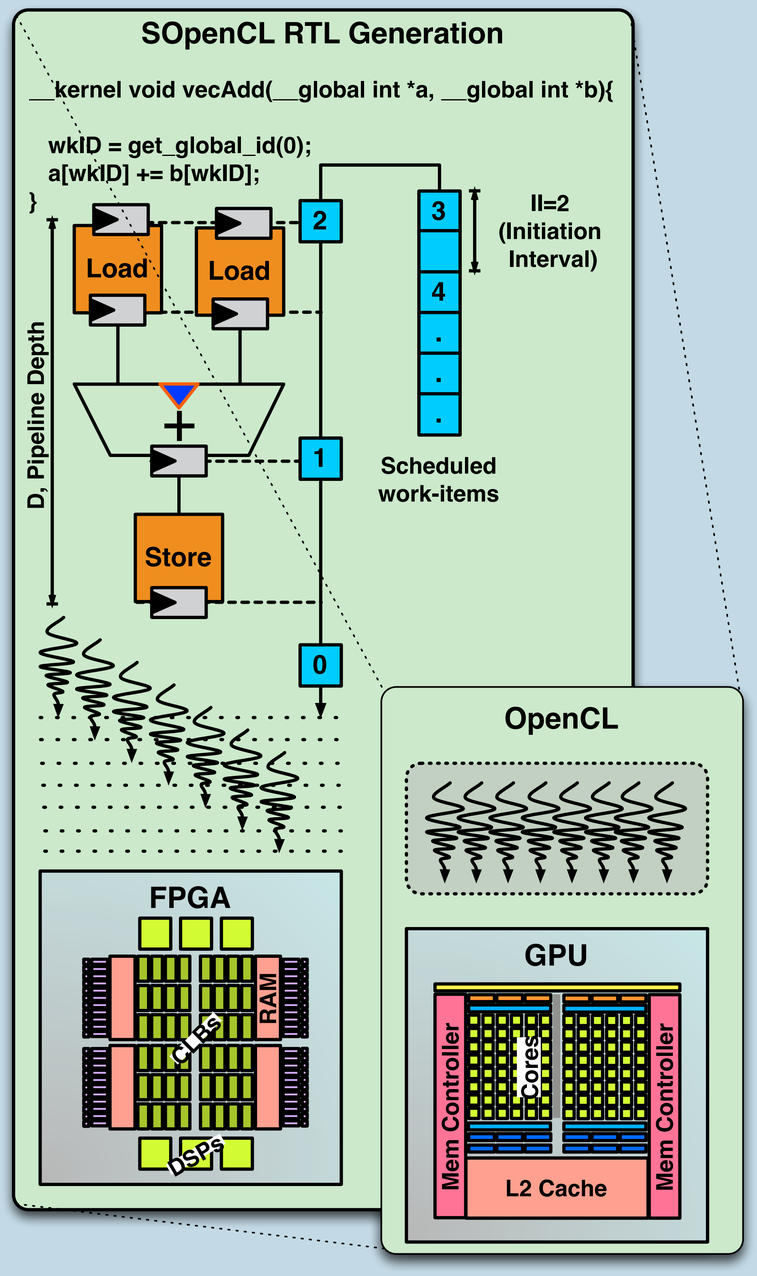
[1]Instituto de Telecomunicações, University of Coimbra, Portugal; [2]Dept. of Computer and Communications Engineering, University of Thessaly, Greece; [3]École Polytechnique Fédérale de Lausanne, Switzerland

Program once, run for all

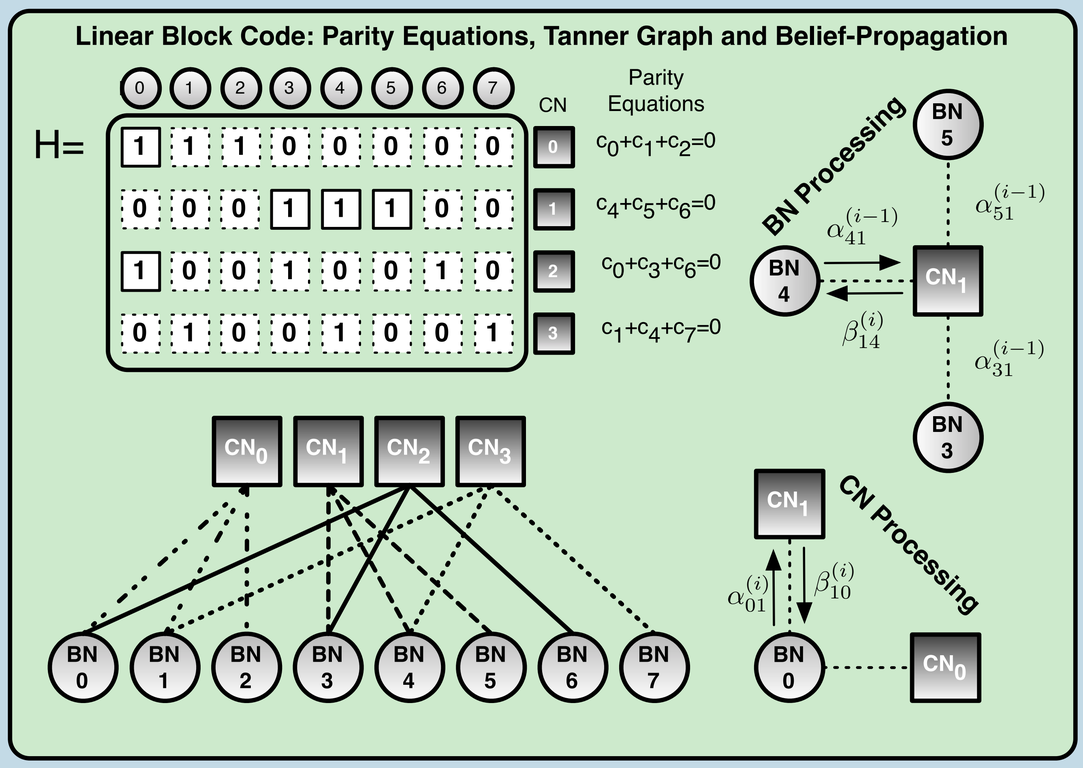
- CPU, GPU, FPGA targets → non-trivial wo/ common model.
- OpenCL compiler → CPUs and GPUs.
- SOpenCL RTL translation [2] → FPGA circuits.



Wide Pipeline on FPGAs



Case Study: Binary LDPC Decoder



Min-Sum Algorithm

Initialization

$$\gamma_n = \frac{2y_n}{\sigma^2} \qquad \alpha_{nm}^{(0)} = \gamma_n$$

CheckNode Processing

$$\beta_{mn}^{(i)} = \prod_{n' \in N(m)\backslash n} \operatorname{sign}\left(\alpha_{n'm}^{(i-1)}\right) \cdot \min_{n' \in N(m)\backslash n} |\alpha_{n'm}^{(i-1)}|$$

BitNode Processing

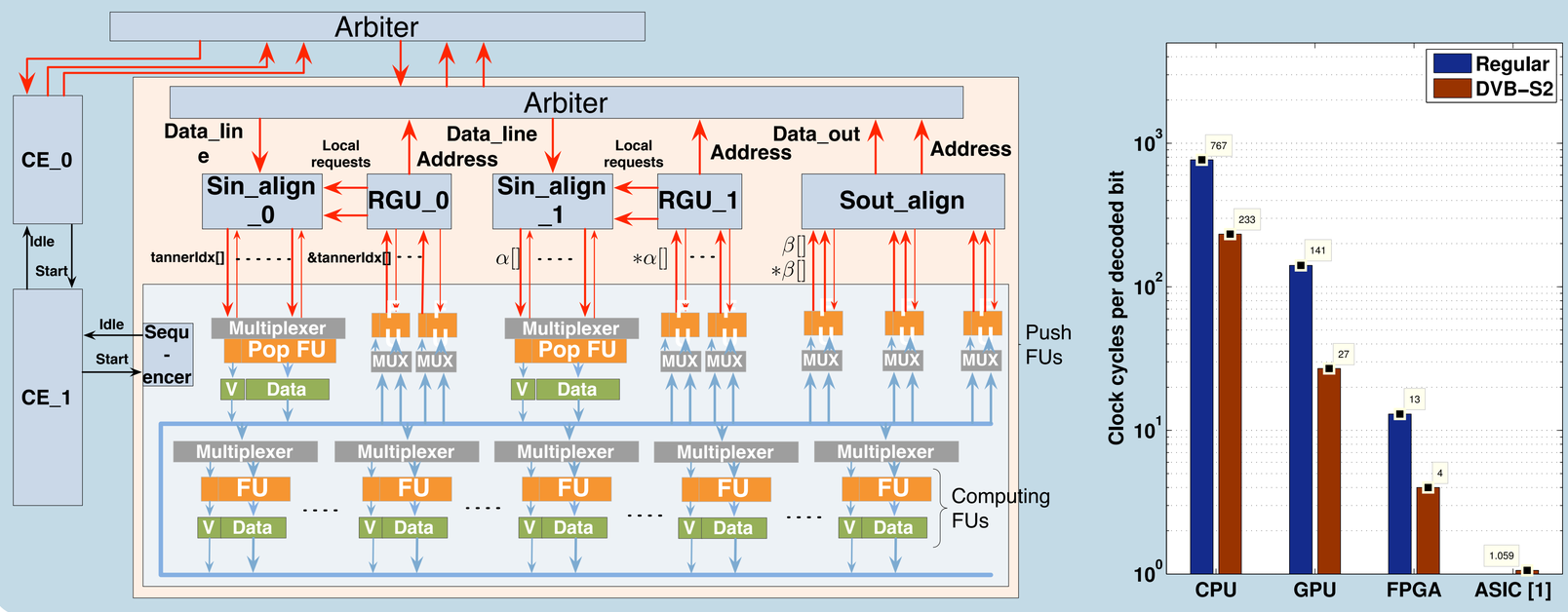
$$\alpha_n^{(i)} = \gamma_n + \sum_{m' \in M(n)\backslash m} \beta_{mn}^{(i)}$$

Hard-decision

$$\alpha_n^{(i)} = \gamma_n + \sum_{m' \in M(n)} \beta_{mn}^{(i)}$$

- (N, M) LDPC code is parallelized in $\{N, M\}$ work-items, work-item-per-node [3].

Generated CN Accelerator



Synthesis Details

CN	Slices	FFs	LUTs	Freq. [MHz]	D [cycles]	BN	Slices	FFs	LUTs	Freq. [MHz]	D [cycles]
II=1	11600	41892	38759	101	102	II=1	6466	19584	18433	163	53
II=2	21424	54872	81526	97	103	II=2	6201	18246	17957	176	54
II=8	27556	57582	58788	53	210	II=8	6747	16791	17983	168	109

- Virtex6 LX760 FPGA slice occupancy $\{15.2\%, 23.3\%, 54.3\%\}$ for II=$\{1, 2, 8\}$.

References

[1] Frank Kienle et al.: *A synthesizable IP Core for DVB-S2 LDPC Code Decoding*, IEEE DATE'05

[2] Owaida et al.: *Shortening design time through multiplatform simulations with a portable OpenCL golden-model: the LDPC decoder case.* IEEE FCCM'11

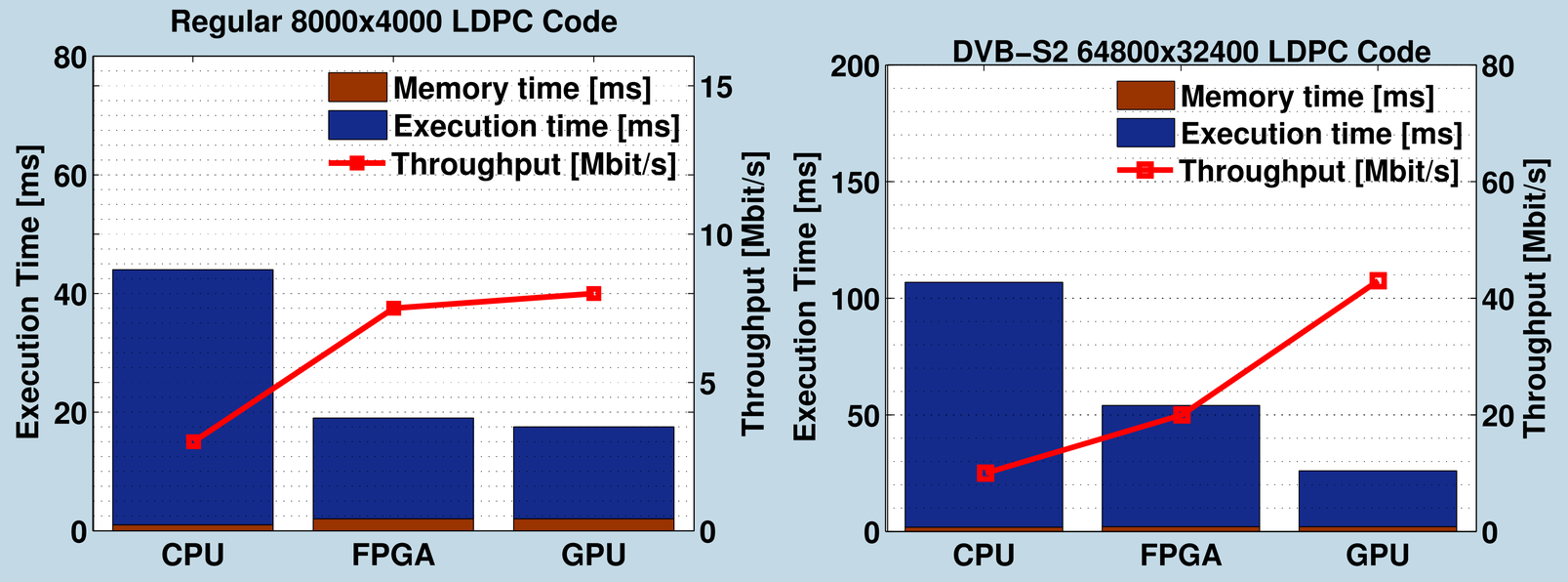
[3] Falcao et al.: *Portable LDPC Decoding on Multicores using OpenCL.* IEEE Signal Processing Magazine, 2012

Acknowledgements

This work was supported by the Fundação para a Ciência e Tecnologia (FCT) grants PEst-OE/EEI/LA0008/2011 and SFRH/BD/78238/2011.

Experimental Results: Throughput and Latency

- Platforms: AMD Phenom 945 X II, Nvidia Tesla C2050, Virtex6 LX760



- Execution time = no. work-items × II + D [cycles] ⇐ no. work-items ≫ D the time is dominated by no. work-items × II → best throughput/latency case: II = 1 clock cycle.

Conclusions and Future Work

- SOpenCL allows fast FPGA prototyping, using the OpenCL programming model. The accelerator efficiency is substantially higher than general purpose platforms and 10x lower than the ASIC implementation [1].
- Massively parallel case study → the no. work-items dominated the execution time. - Accelerator performance vs. data granularity should be considered.