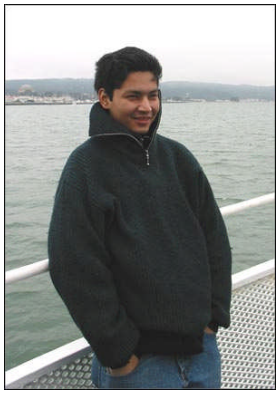
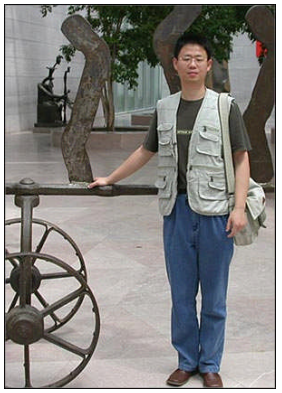
Implementation of Elliptic Curve Cryptosystems over GF(2^n) in Optimal Normal Basis on a Reconfigurable Computer



Sashisu Bajracharya, Chang Shu, Kris Gaj

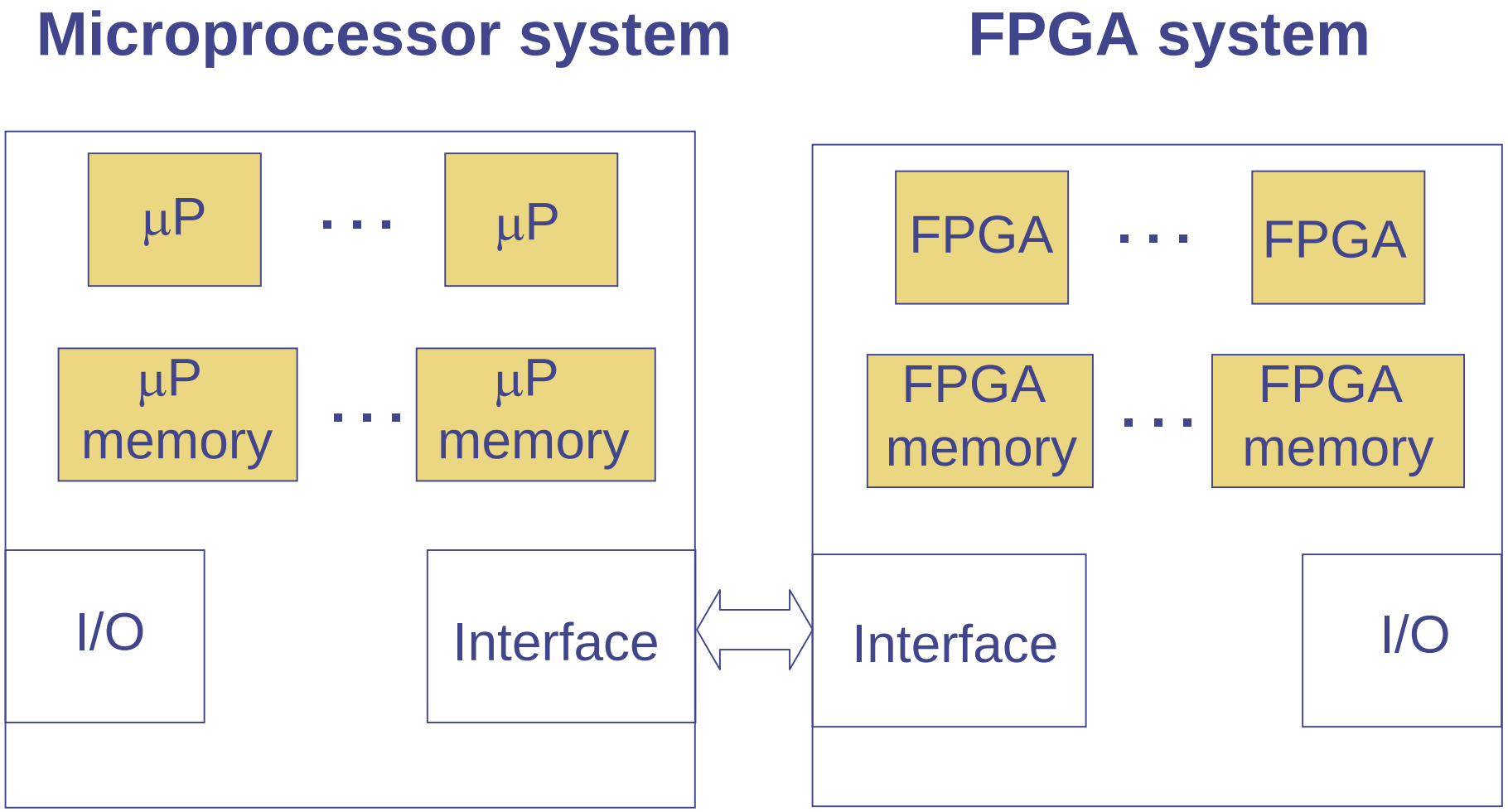
George Mason University



Tarek El-Ghazawi

The George Washington University

What is a reconfigurable computer?



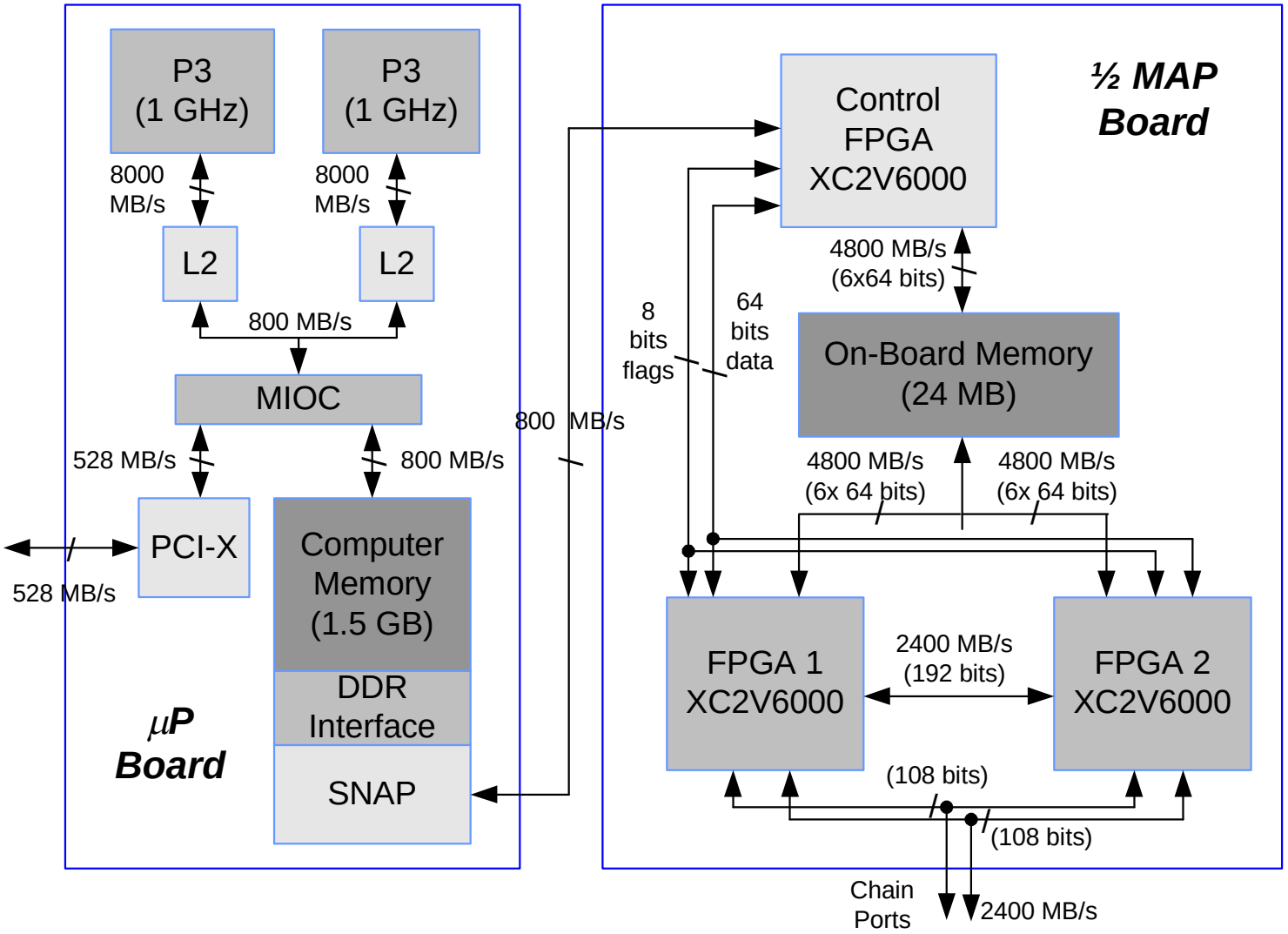
Why cryptography is a good application for reconfigurable computers?

- computationally intensive arithmetic operations
- unconventionally long operand sizes (160-2048 bits)
- multiple algorithms, parameters, key sizes, and architectures = need for reconfiguration

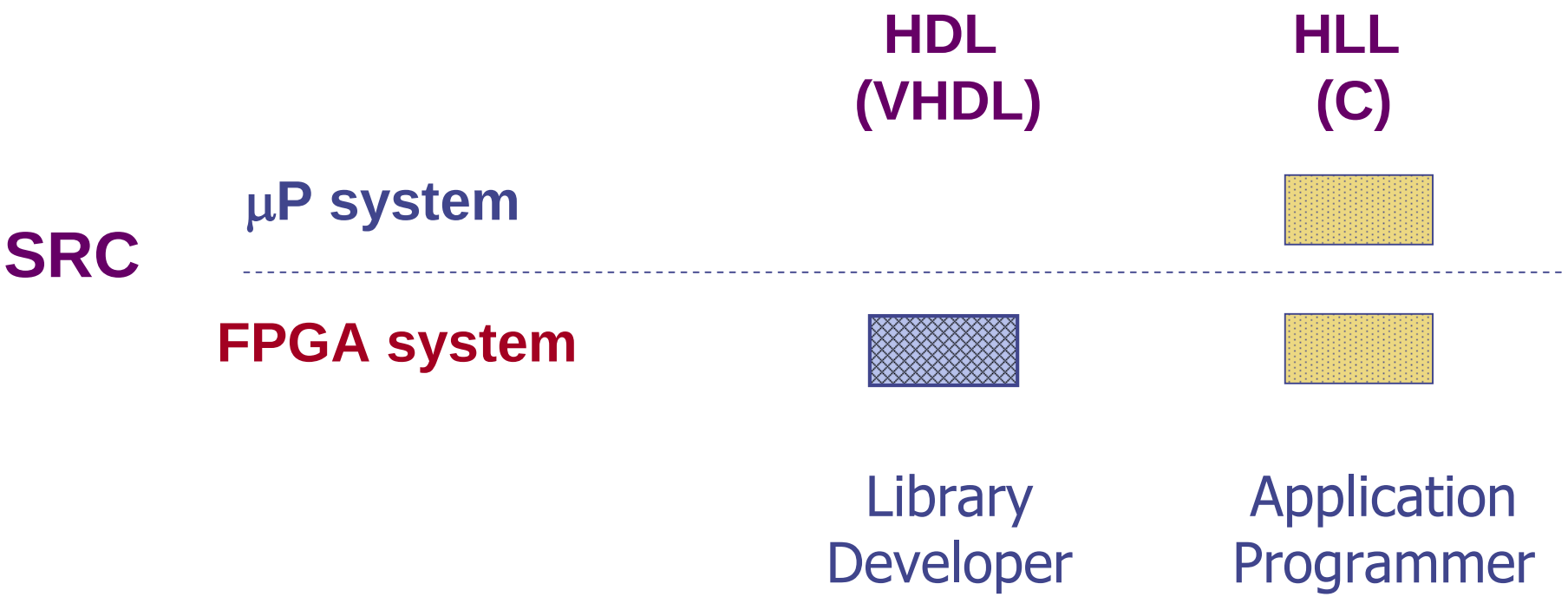

SRC
Reconfigurable Computer

SRC-6E from SRC Computers, Inc.

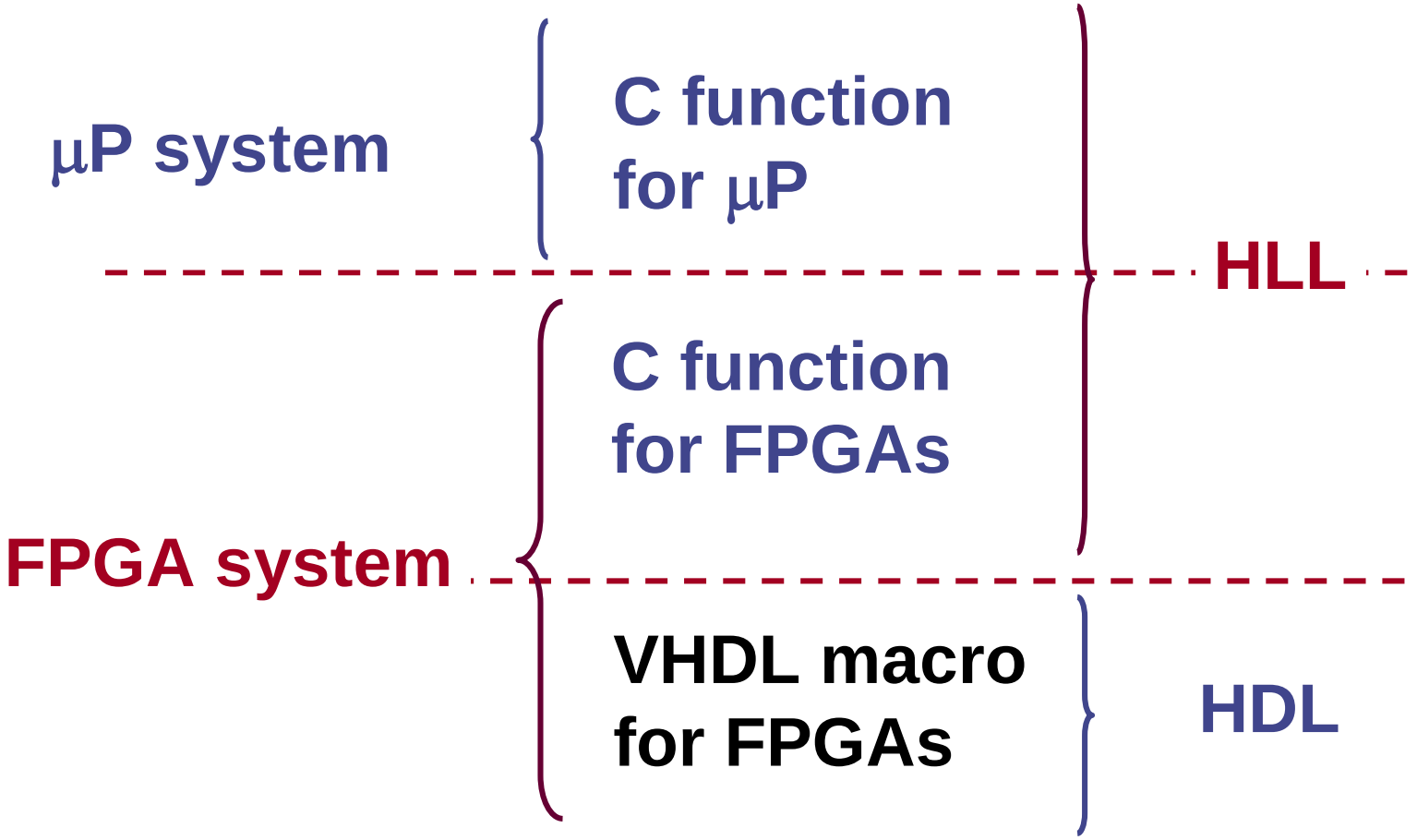
SRC Hardware Architecture

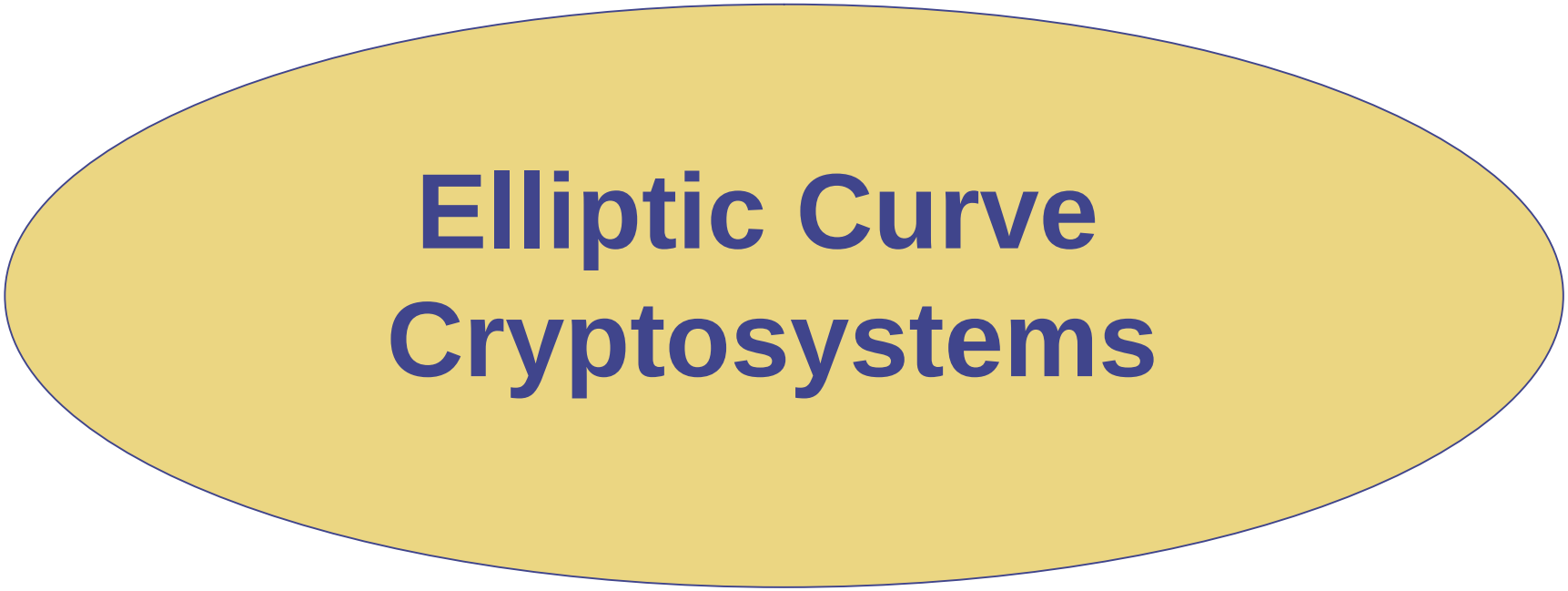


SRC Programming



SRC Program Partitioning

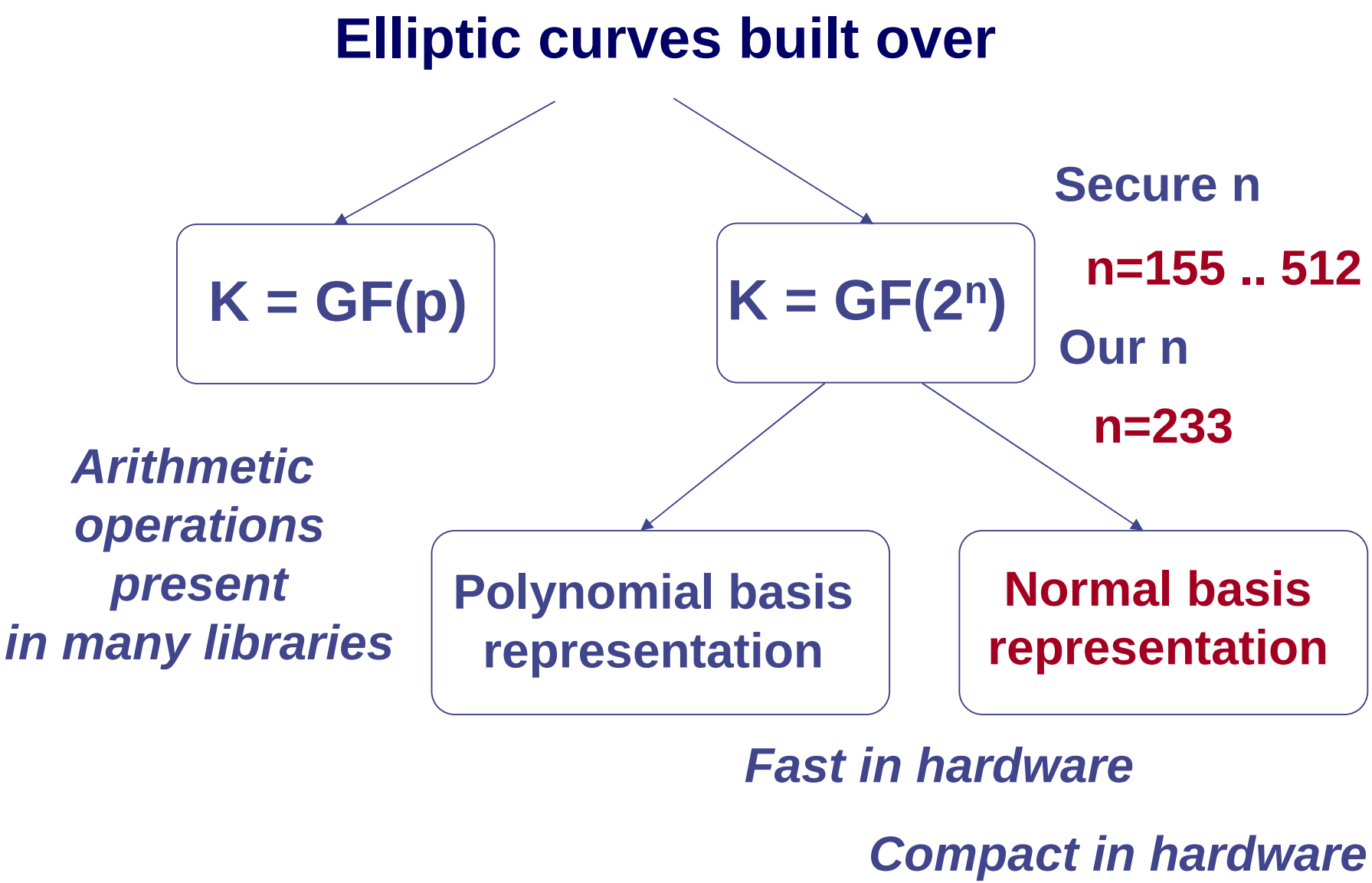



Elliptic Curve
Cryptosystems

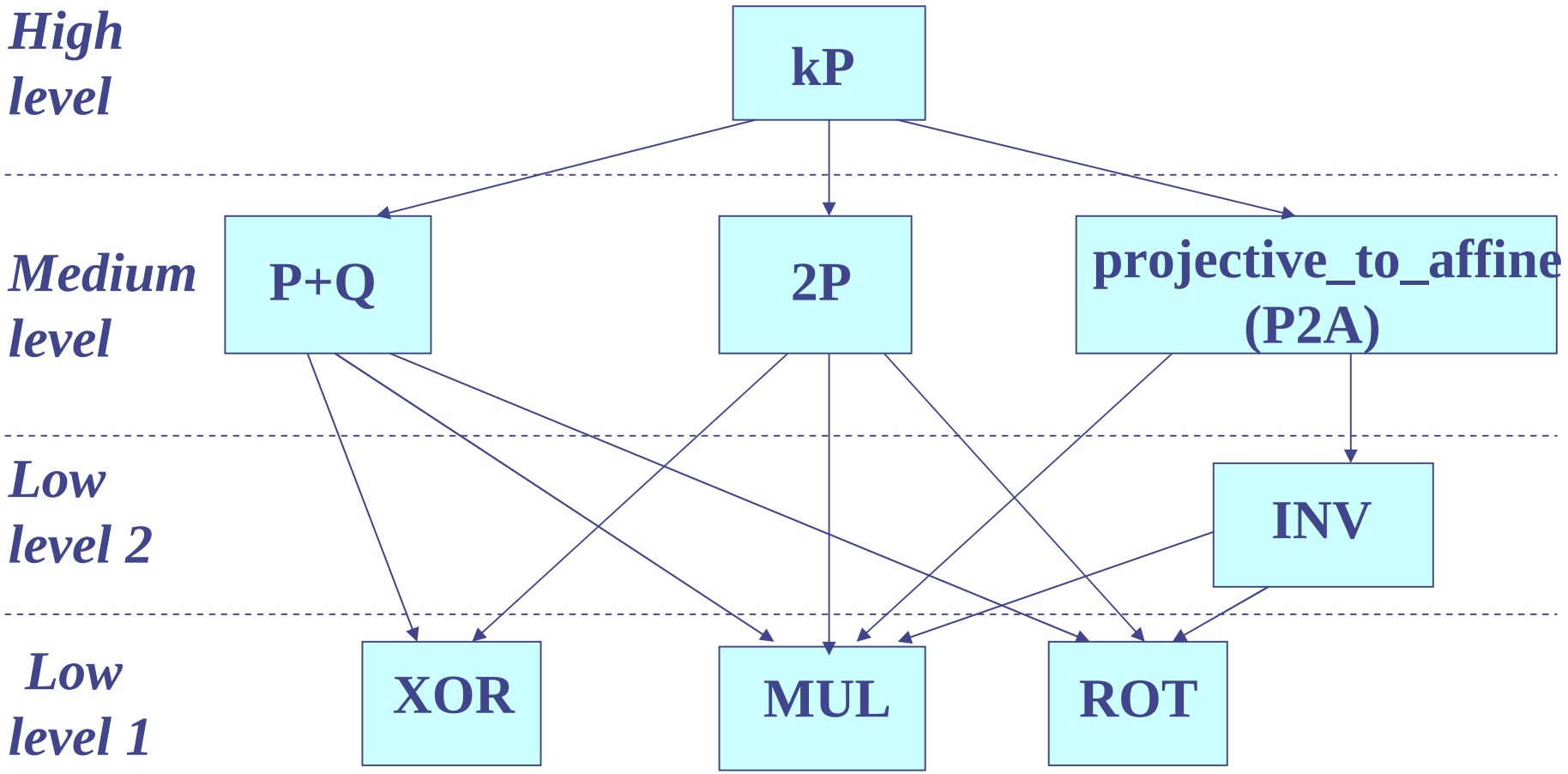
Elliptic Curve Cryptosystems (ECC)

- ✓ a family of cryptosystems, rather than a single cryptosystem = added security but need for reconfiguration
- ✓ public key (asymmetric) cryptosystems used for key agreement and digital signatures
- ✓ implementations must be optimized for minimum latency rather than maximum throughput = limited speed-up from parallel processing

Three Families of Elliptic Curves



Hierarchy of functions



Investigated Partitioning Schemes

µP Software Only

C function
for µP

kP

C function
for FPGA

VHDL
macro

Based on public-domain code by Rosing M.,
Implementing Elliptic Curve Cryptography,
Manning, 1999

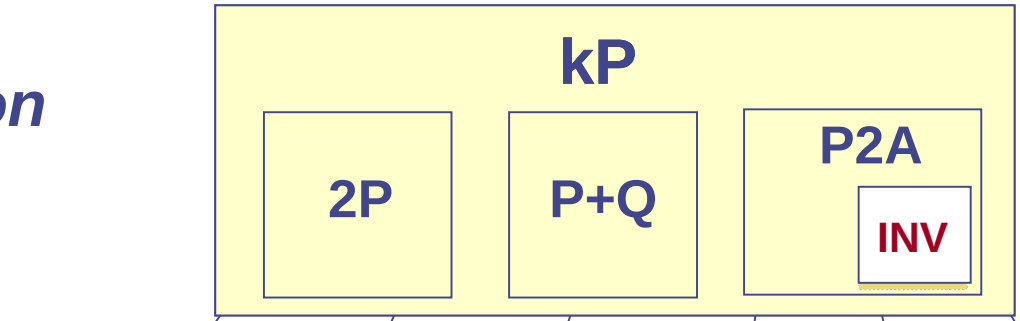
0HL1 Partitioning

C function for µP

0

C function for FPGA

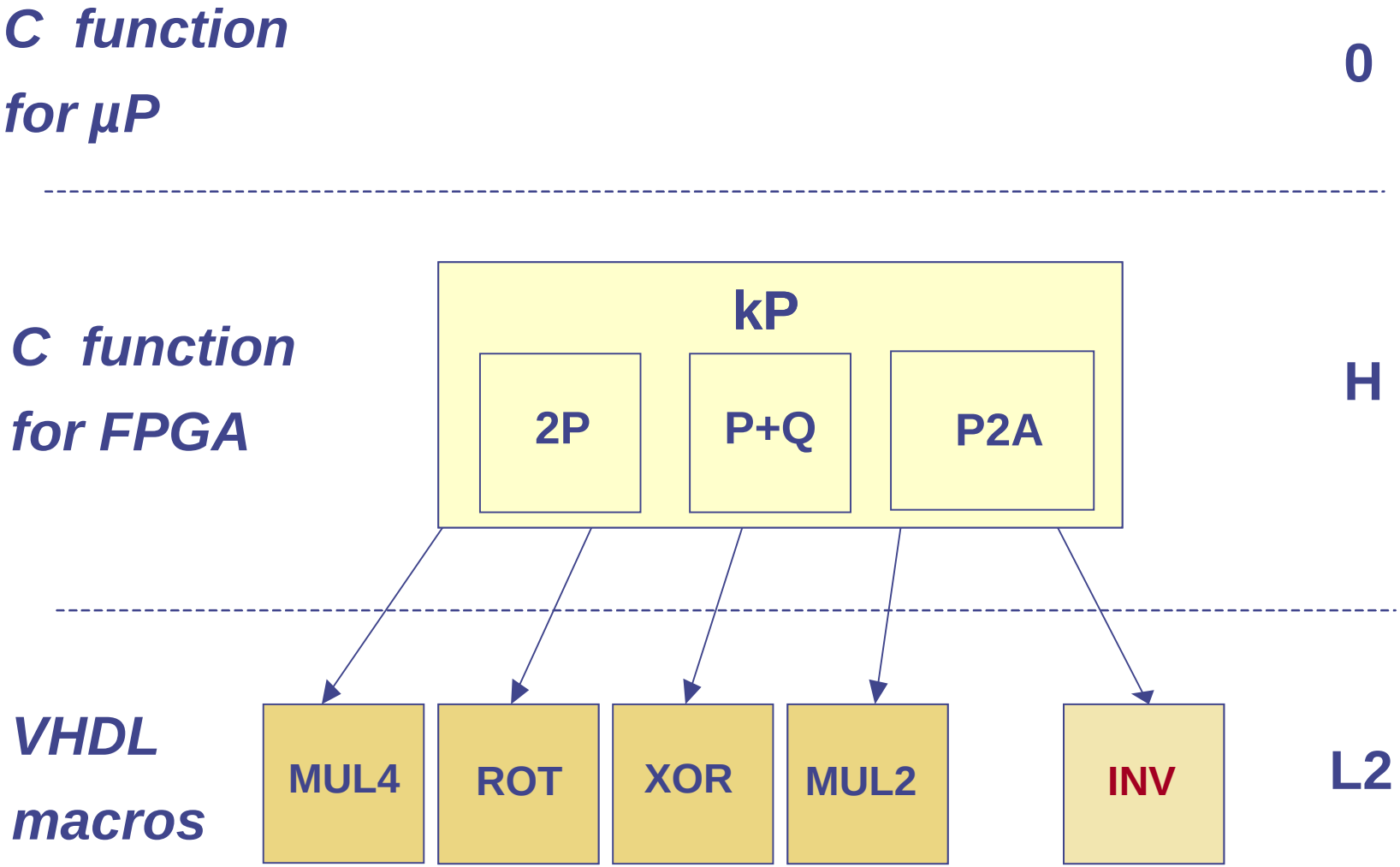
H



VHDL macros

L1

0HL2 Partitioning

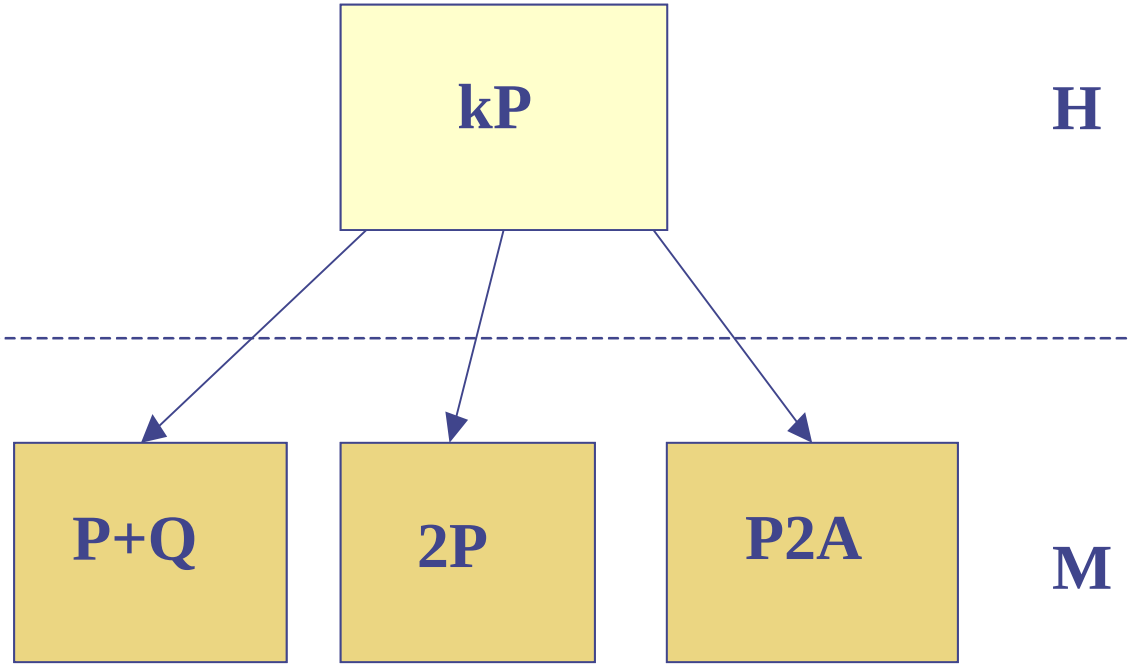


0HM Partitioning

C function for µP

0

C function for FPGA



VHDL macros

00H Partitioning (VHDL only)

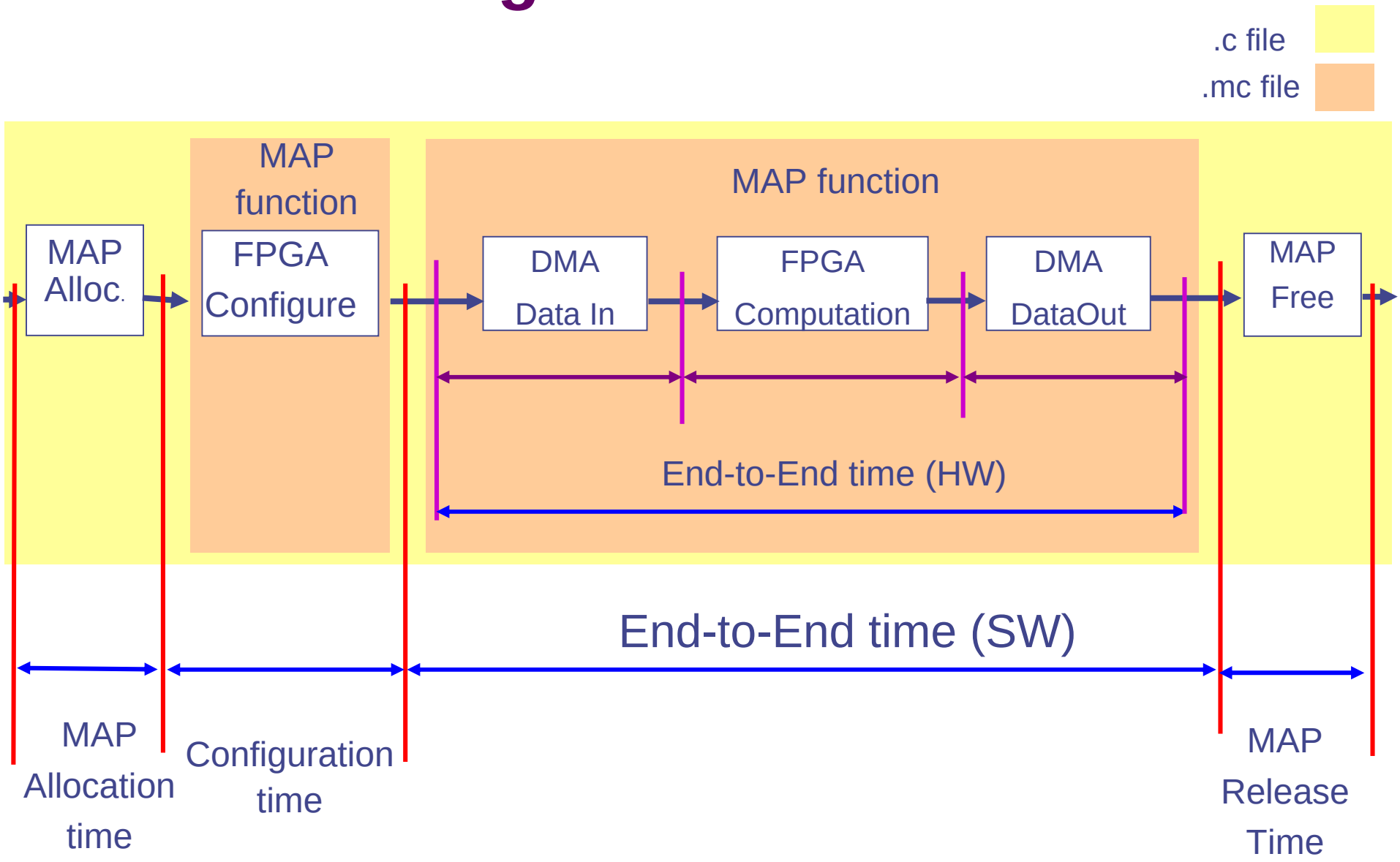
C function for μP — 0

C function for FPGA — 0

VHDL macro — kP — H

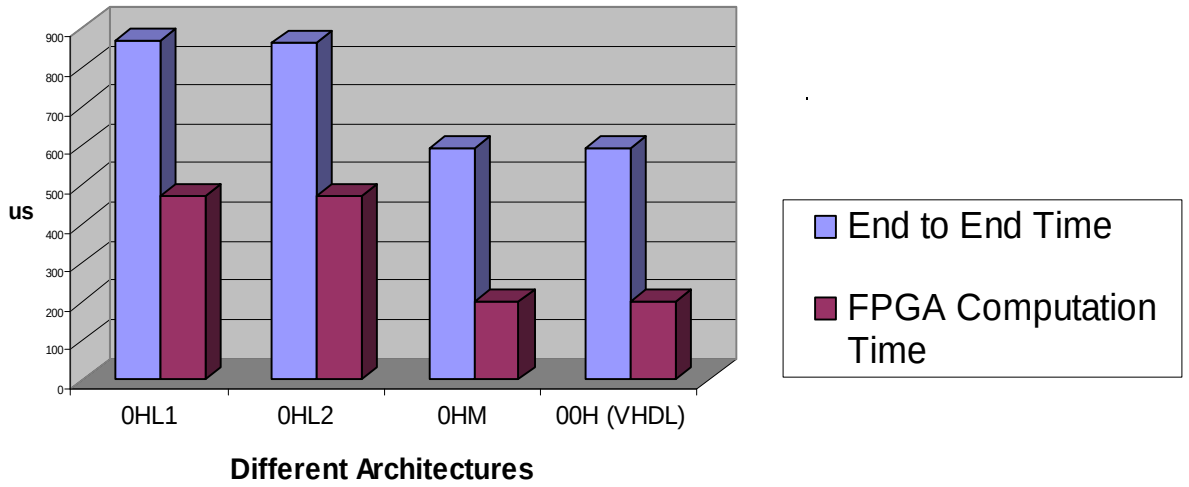

Results

Timing Measurements



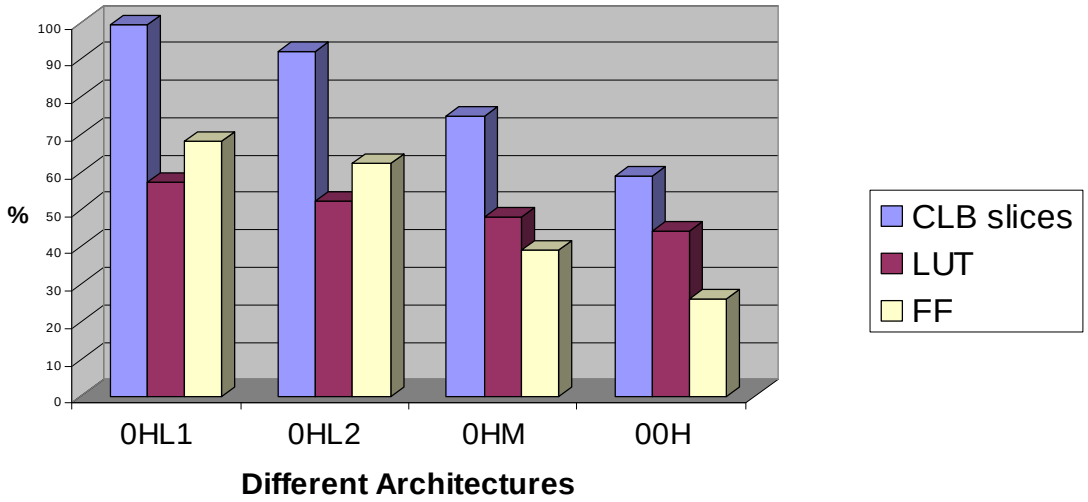
Results (Latency)

System Level Architecture	End-to-End Time (µs)	Data Transfer In Time(µs)	FPGA Compu-tation Time (µs)	Data Transfer Out Time (µs)	Total Overhead (µs)	Speedup vs. Software	Slow-down vs. VHDL macro
Software	772,519						
0HL1	866	37	472	14	394	**893**	**1.46**
0HL2	863	37	469	14	394	**895**	**1.45**
0HM	592	37	201	12	391	**1305**	**1**
VHDL macro	592	39	201	17	391	**1305**	**1**



Results (Area)

System Level Architecture	% of CLB slices (out of 33792)	CLB increase vs. pure VHDL	% of LUTs (out of 67,584)	LUT increase vs. pure VHDL	% of FFs (out of 67,584)	FF count increase vs. pure VHDL
Software	N/A					
0HL1	99	**1.68**	57	**1.3**	68	**2.61**
0HL2	92	**1.56**	52	**1.18**	62	**2.38**
0HM	75	**1.27**	48	**1.09**	39	**1.5**
00H	59	**1**	44	**1**	26	**1**



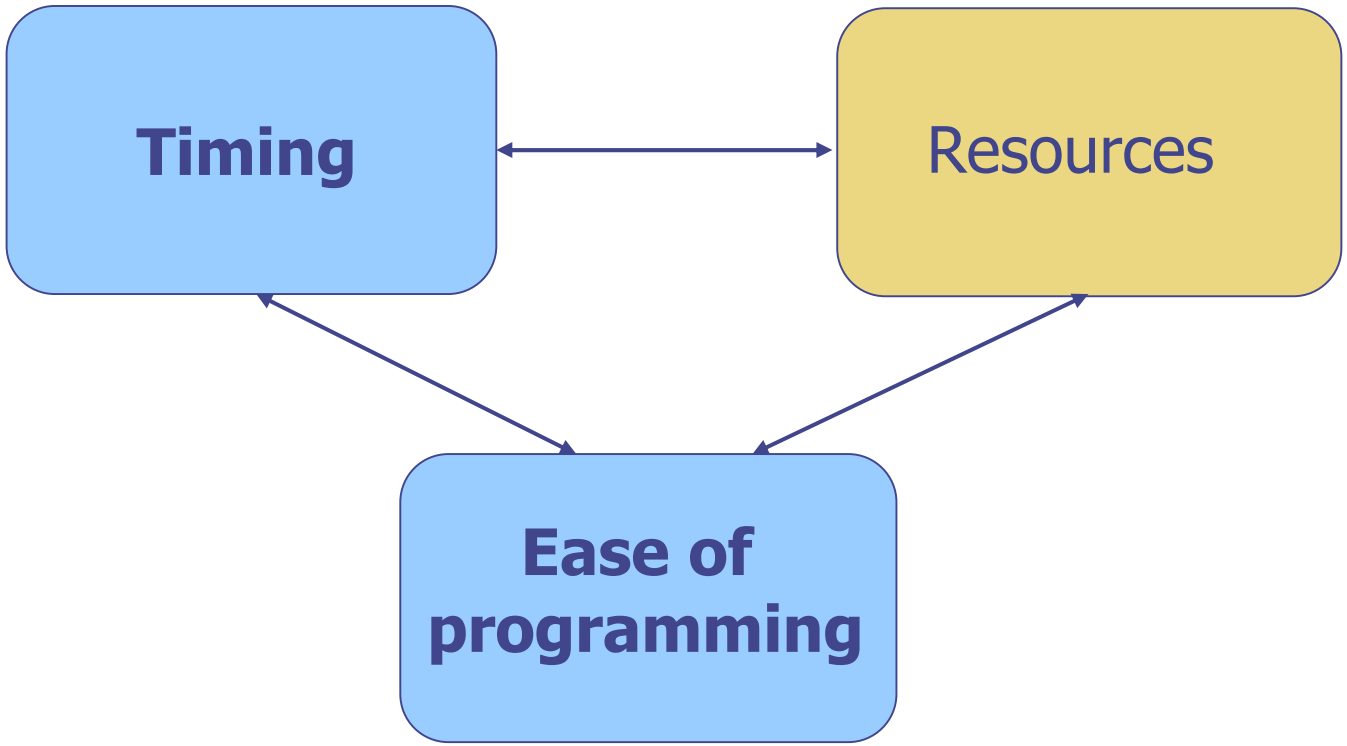
Number of lines of code

Algorithm Partitioning Scheme	VHDL	Macro Wrapper	MAP C	Main C
0HL1	**1007**	260	**371**	153
0HL2	1291	230	349	153
0HM	1744	160	185	153
VHDL macro	**1960**	36	**78**	153

Conclusions

Conclusions

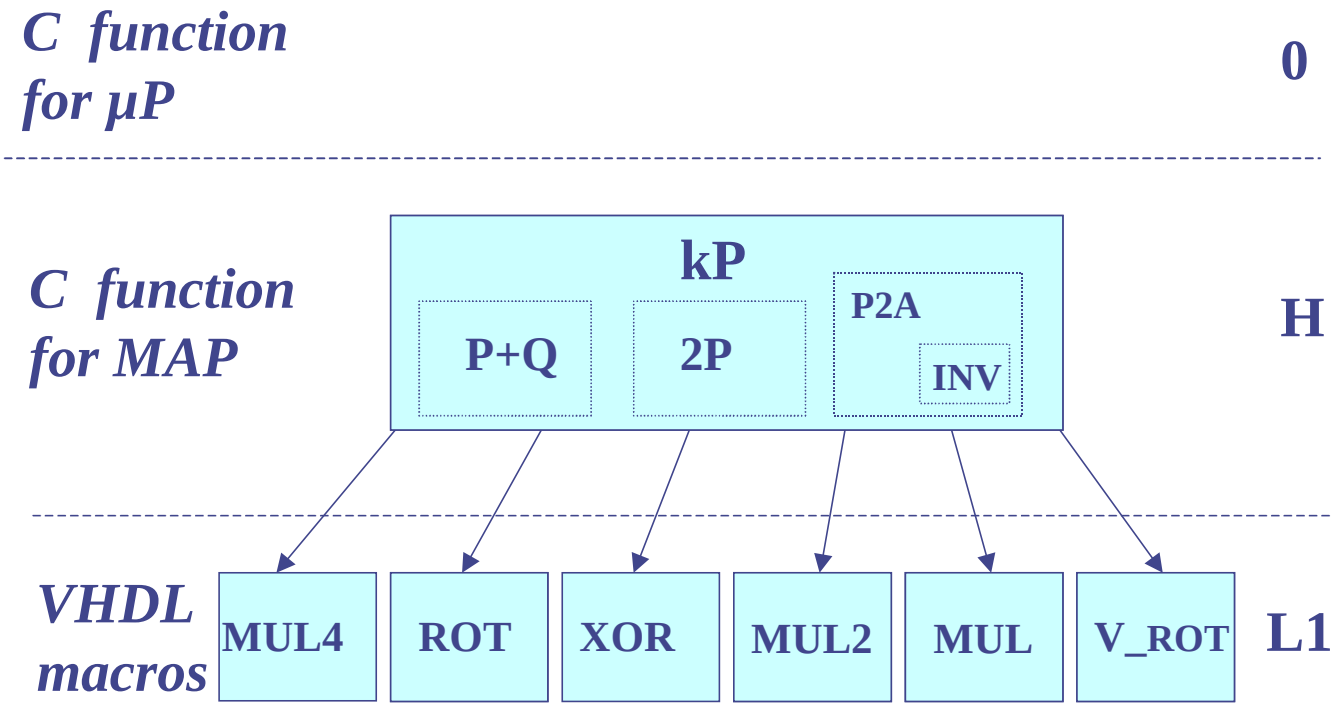
Assuming focus on:



Conclusions – cont.

The best implementation approach:

0HL1 partitioning scheme



893 speedup vs. software and only 0.46 times slowdown versus pure VHDL with ease of implementation

Conclusions

- Elliptic Curve Cryptosystem implementation challenging for reconfigurable computers because of
  - optimization for latency rather than throughput
  - limited amount of parallelism
- First publication showing a 1000x speed-up for a reconfigurable computer application optimized for data latency