

Floating Point FPGAs

Philip Leong
phwl@doc.ic.ac.uk

Imperial College London

Overview

- C-FPGAs vs uPs
- Floating point FPGAs
- Virtual embedded blocks

Overview

- C-FPGAs vs uPs
- Floating point FPGAs
- Virtual embedded blocks

High Performance Applications

- C-FPGAs
 - Signal processing, cryptography, networking, string matching
- Microprocessors
 - DSP, linear systems, differential equations, optimisation, simulation

C-FPGAs vs uPs

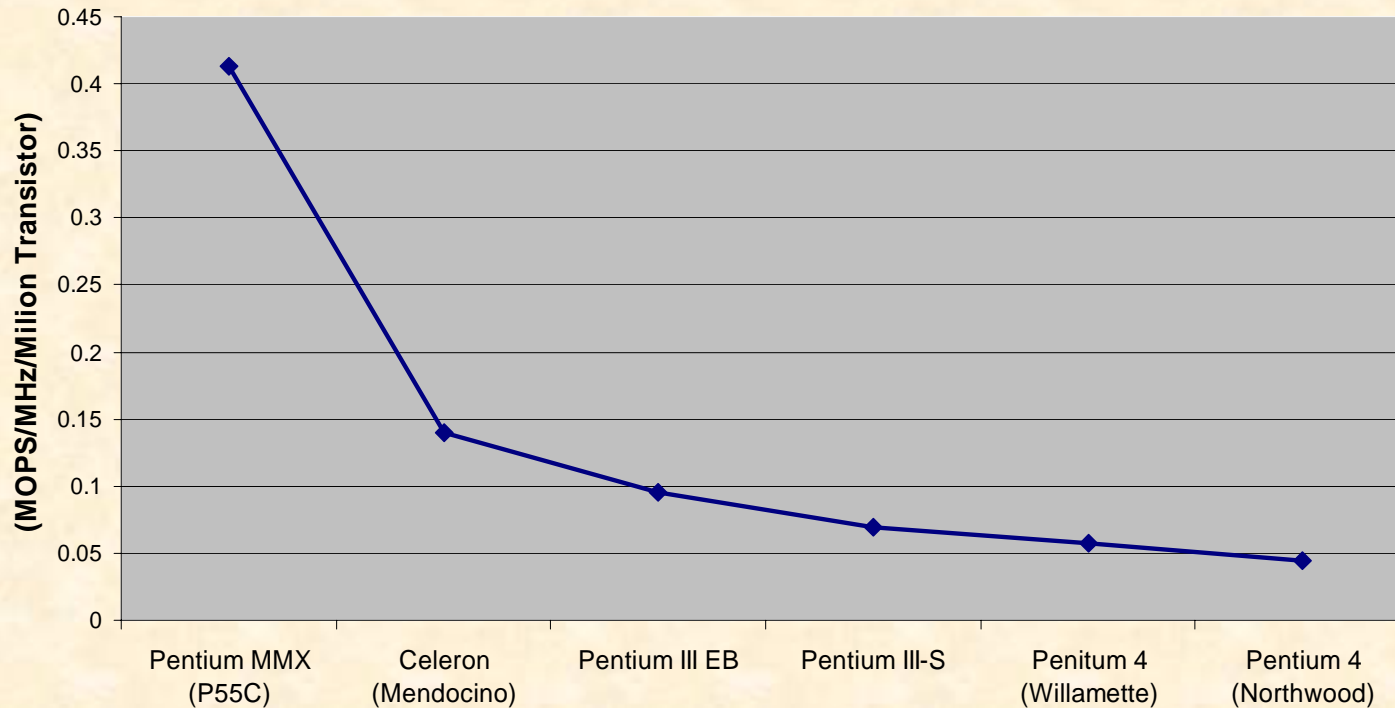
- Strengths

- More parallelism
- Higher computational density
- Lower power consumption
- Higher memory bandwidth, direct control of accesses
- Can be fault tolerant

- Weaknesses

- Long wordlengths
- Floating point
- Low clock frequency
- Run out of resources
- Design time
- Legacy code

uP Computational Density



- Problems with current microprocessors
 - Serial instruction stream limits parallelism
 - Power consumption limits performance
 - Memory bandwidth limits density and performance

Intel Pentium 5 Prescott

Trace Cache Access, next Address Predict

Trace Cache Branch Prediction Table (BTB), 1024 entries.
Return Stacks (4 x 16 entries)
Trace Cache next IPs (4x)

Instruction Decoder

Up to 4 decoded uOps/cycle out (from max. one x86 instr/cycle)
Instructions with more than four are handled by Micro Sequencer
Raw Instruction Bytes in Data TLB, 64 entry fully associative, between threads dual ported (for loads and stores)
Front End Branch Prediction Tables (BTB), shared, 4096 entries in total

Instruction TLB's 128 entry, fully associative for 4k and 4M pages. In: Virtual address [47:12]
Out: Physical address [39:12] + 2 page level bits

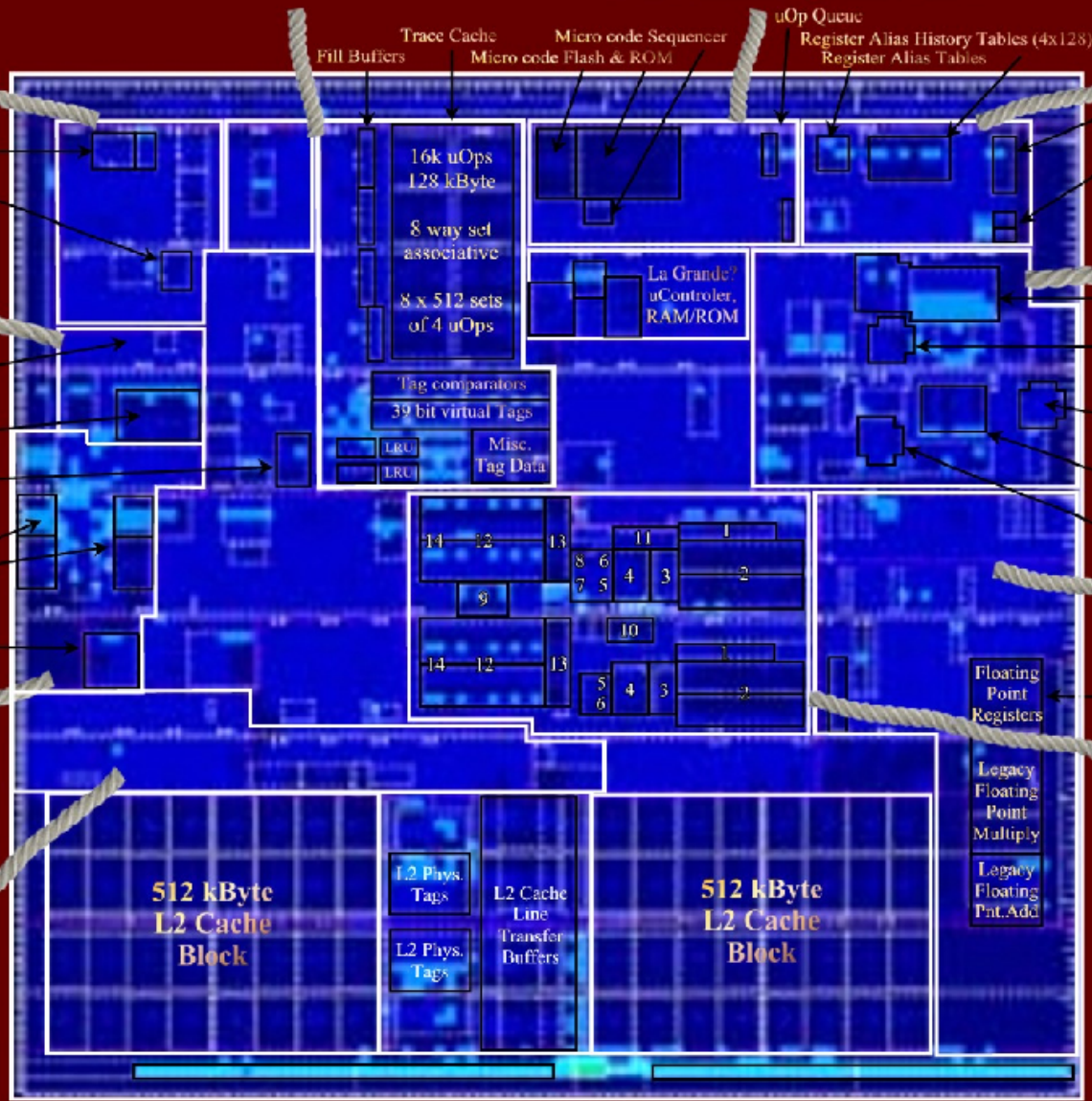
Instruction Fetch from L2 cache and Branch Prediction

Front Side Bus Interface, 533..800 MHz

Instruction Trace Cache

Execution Pipeline Start

Buffer Allocation & Register Rename



Instruction Queue (for less critical fields of the uOps)
General Instruction Address Queue & Memory Instruction Address Queue (queues register entries and latency fields of the uOps for scheduling)

uOp Schedulers

Parallel (Matrix) Scheduler for the two double pumped ALU's
General Floating Point and Slow Integer Scheduler: (8x8 dependency matrix)
FP Move Scheduler: (8x8 dependency matrix)
Load / Store Linear Address Collision History Table
Load / Store uOp Scheduler: (8x8 dependency matrix)

FP, MMX, SSE1..3

Floating Point, MMX, SSE1..3 Renamed Register File
256 entries of 128 bit

Integer Execution Core

- (1) uOp Dispatch unit & Replay Buffer Dispatches up to 6 uOps / cycle
- (2) Integer Renamed Register File 256 entries of 32 bit (+ 6 status flags) 12 read ports and six write ports
- (3) Databus switch & Bypasses to and from the Integer Register File.
- (4) Flags, Write Back
- (5) Double Pumped ALU 0
- (6) Double Pumped ALU 1
- (7) Load Address Generator Unit
- (8) Store Address Generator Unit
- (9) Load Buffer (96 entries)
- (10) Store Buffer (48 entries)

(13) Databus multiplexing
(14) Cache Line Read / Write Transferbuffers and 256 bit wide bus to and from L2 cache
(11) ROB Reorder Buffer 4x64 entries
(12) 16 kByte Level 1 Data cache four way set associative. 1R/1W

Overview

- C-FPGAs vs uPs
- Floating point FPGAs
- Virtual embedded blocks

Floating Point FPGA (FP-FPGA)

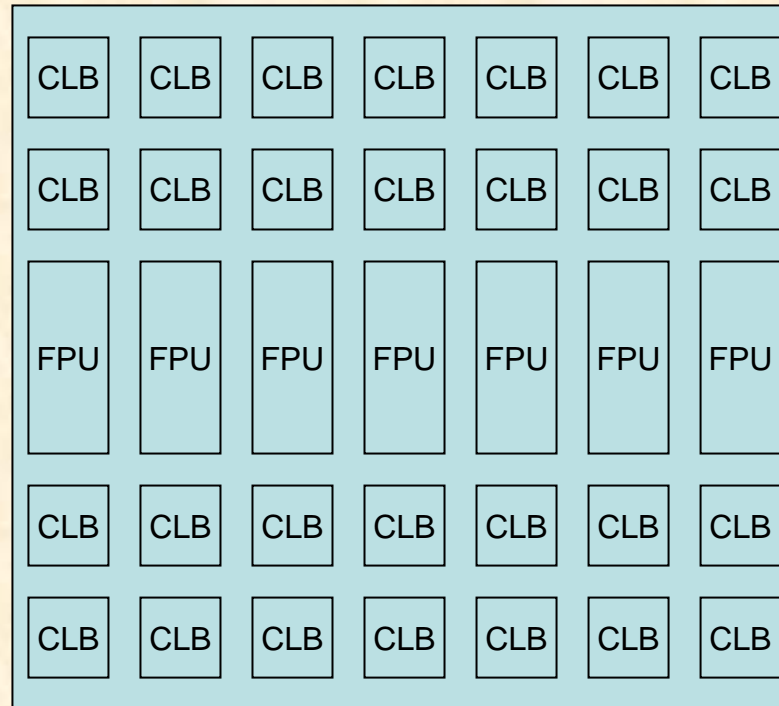
- Weaknesses
 - Long wordlengths
 - Floating point
 - Low clock frequency
 - Run out of resources
 - Design time
- Can we develop an FPGA specifically optimised for floating point applications?
 - Coarse grained architecture
 - Hardwired FPUs
 - Runtime reconfiguration
 - Compilers
- Advantages
 - More transistors used in parallel FPUs than a uP
 - Better floating point performance than standard FPGA/uP
 - Development time reduced as designers do not need to deal with fixed point quantisation issues
 - External memory often bottleneck, FPGAs offer potentially higher bandwidth (multiple channels) as well as custom control of cache
 - Branch mispredictions don't cause tens of cycles to recover

Potential Applications

- Scientific computing and embedded systems
- Areas
 - Signal processing
 - CAD
 - Molecular dynamics, N-body problem
 - Differential equations
 - Linear systems
 - Financial engineering
 - Optimisation
 - Any computationally intensive floating point problem
- Specific programs to accelerate
 - Linpack (solving a system of linear equations, supercomputers are ranked by this benchmark)
 - Spice (generation of matrix, LU decomposition of sparse matrix)
 - N-body problem

An Initial Architecture

- Island style FPGA + floating point units + memory



- What sort of speedup could we expect?

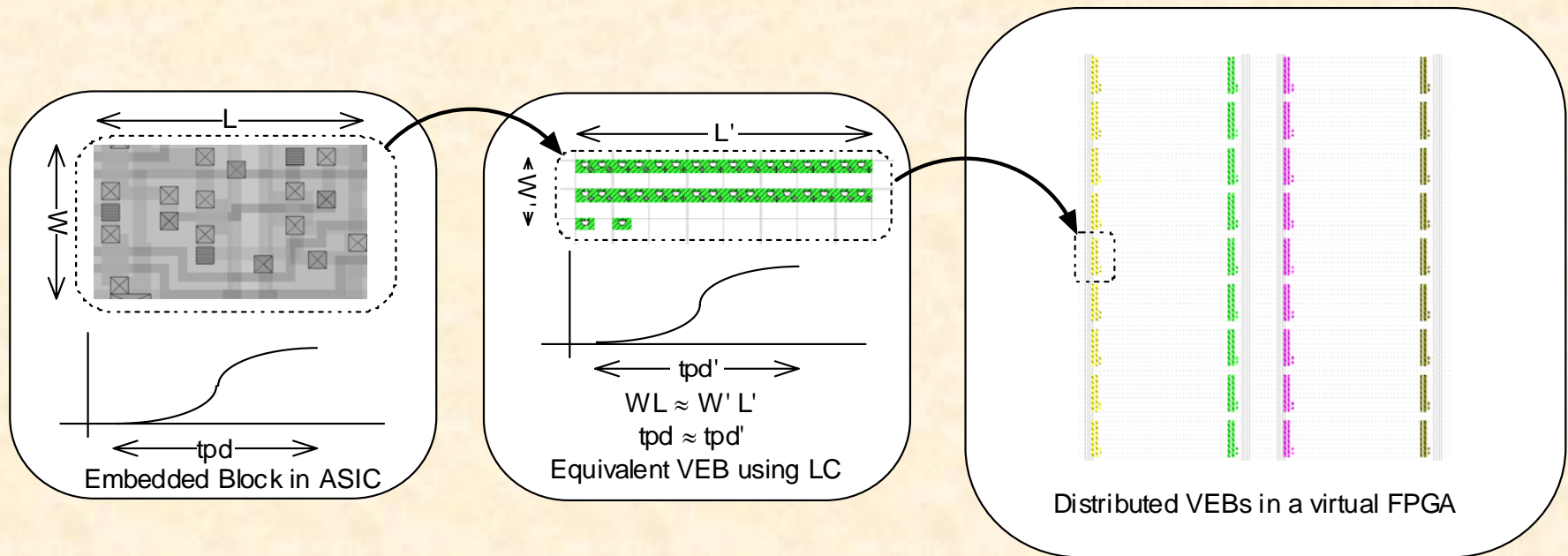
Overview

- C-FPGAs vs uPs
- Floating point FPGAs
- Virtual embedded blocks

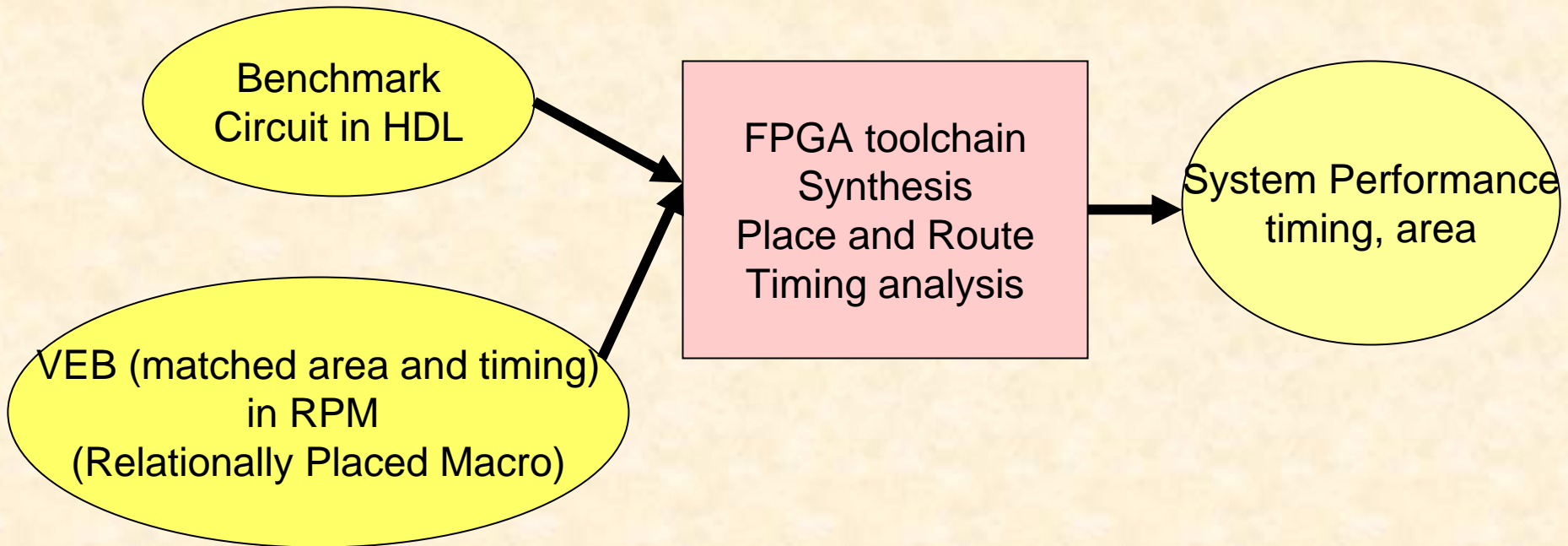
Virtual Embedded Blocks

- Use existing tools to be used to study the effects of embedded elements in FPGAs
- Evaluate accuracy by modelling existing embedded elements in FPGAs over various applications.
- Explore technology trends based on systematic variation of VEB parameters in applications.

VEB design flow (generic)

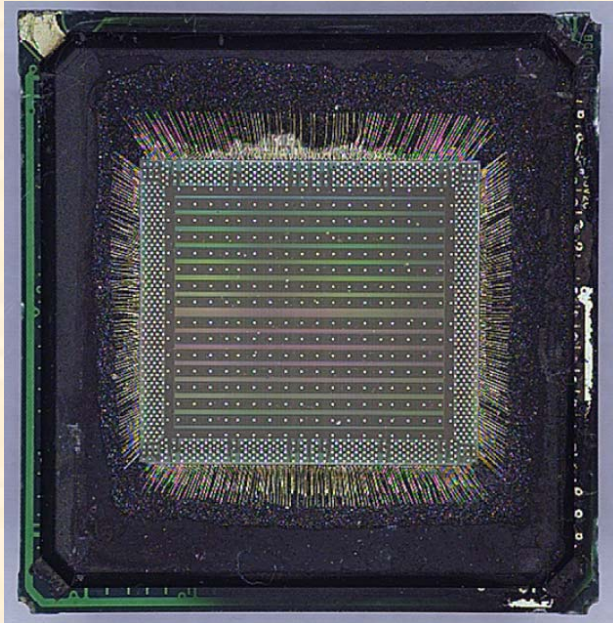


VEB design flow



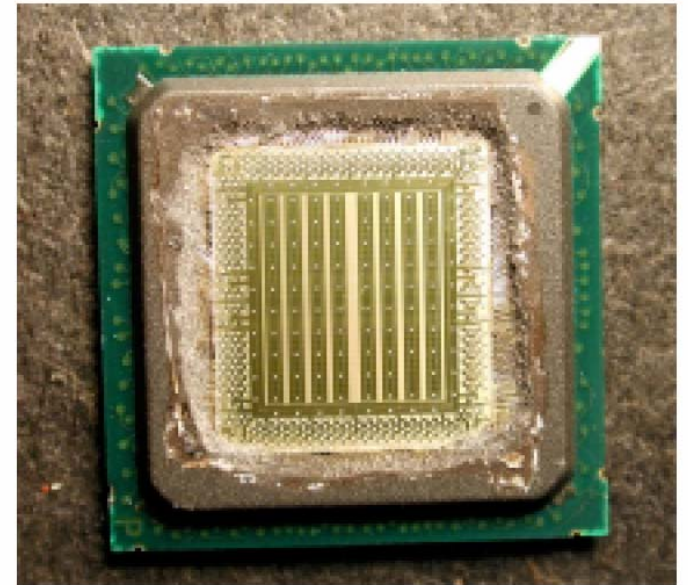
Area Model

- Use logic cells to model real ASIC embedded blocks
- Estimate LC area (normalised to feature size) from die photos



Xilinx Virtex II XQR2V3000

(1.5um, 8 metal, 16x16mm)



Xilinx Virtex II XC2V1000

(1.5um, 8 metal, 9.7x9.7mm)

Area Model

- Area model can be used to compare logic cell area to any embedded block
 - Logic Cell (LC): 442, 000 = 1 LC
 - Multiplier: 2, 751, 000 (normalised) ~ 6 LC
 - Blue-gene floating point unit (FPU) ~ 570 LC

Delay Model

- Match delays using LC
 - Use adder carry chain to model the delay
- For small blocks, may fail to match both area and delay

Verification of VEB using EM

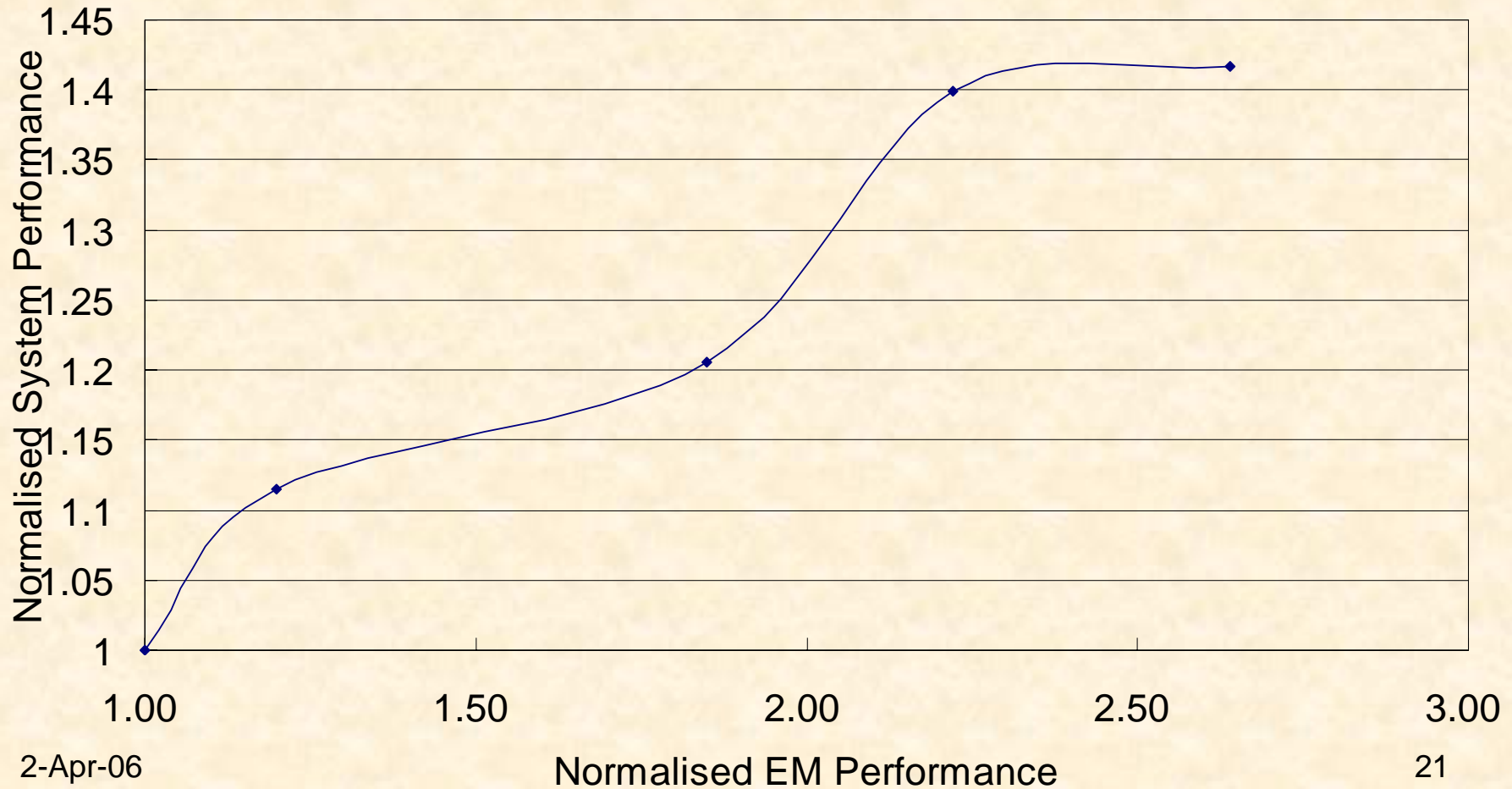
	EM delay (ns)	VEB delay (ns)	Overall Diff (%)
DSCG	4.599	4.981	8
FIR4	4.616	4.794	2
ODE	4.402	4.539	3
MM3	4.859	4.815	1
BFLY	5.668	5.224	8
MUL34	11.191	11.287	1
MUL68	12.553	14.099	11
MUL136	14.632	13.248	10
BGM	14.055	13.866	1
BGM (retimed)	11.594	11.602	0

Difference at most 11%

Faster EMs

- Explore the speedup by increasing the performance of embedded multiplier
 - Tested on fixed-point BGM circuit (bgm)

System performance vs EM Performance (BGM)



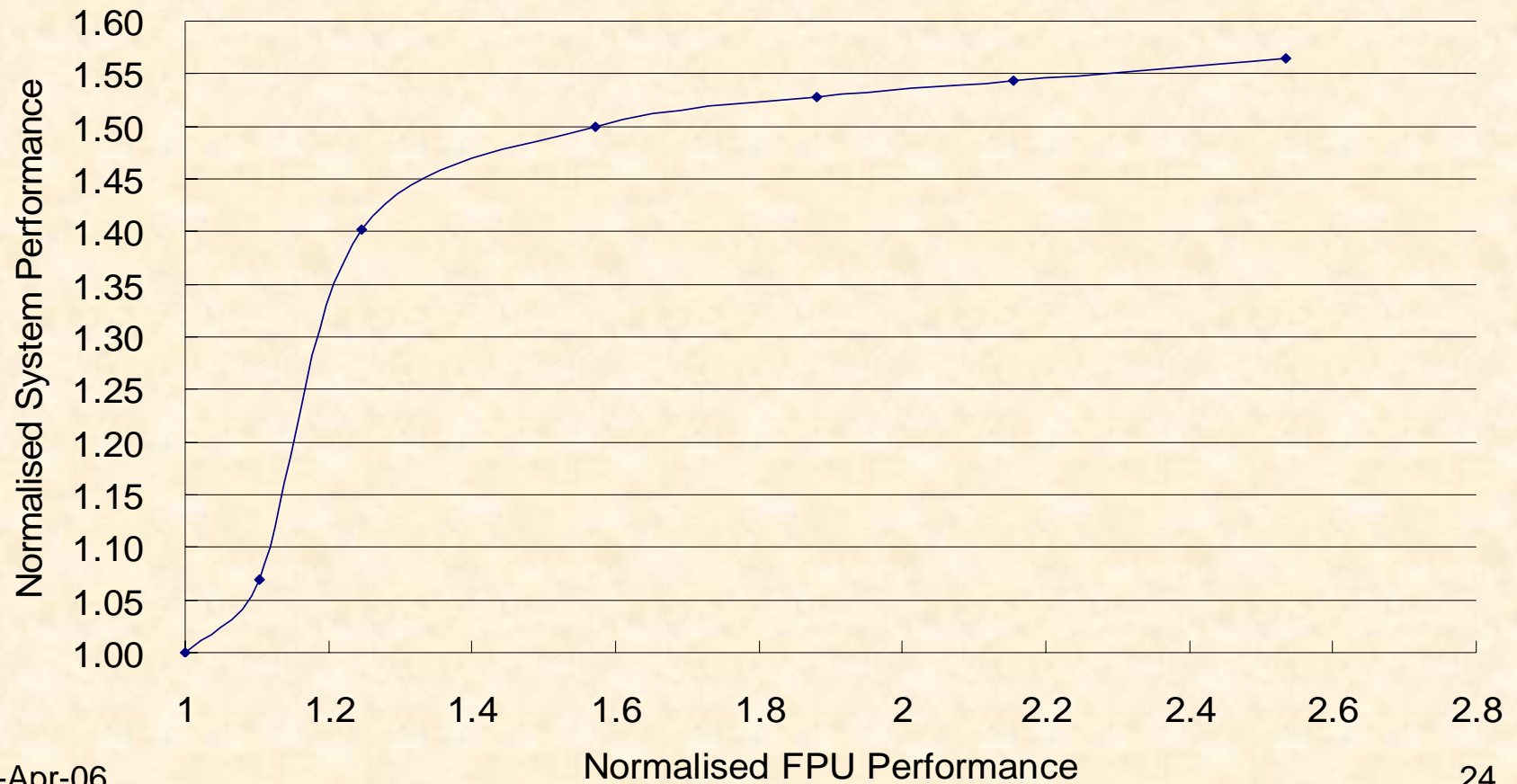
Embedded FPU

- Embedding a floating point unit
 - FPU delay and area based on published Blue Gene data
 - 700MHz, $4.26 \text{ mm}^2 = 570 \text{ LCs}$
 - For FPGAs, reduce latency and clock frequency by a factor of 5: 140 MHz, one cycle latency
- Explore the speedup by increasing the performance of floating point unit
 - Tested on floating-point butterfly circuit (bfly)

System Performance for Different Benchmarks

	FPGA		VEB		Reduction Factor	
	size (LC)	delay (ns)	size (LC)	delay (ns)	size	delay
dscg	19006	22.711	3420 + 940	8.807	4.4	2.6
fir4	20590	23.545	3990 + 996	9.539	4.1	2.5
ode	13984	17.756	2850 + 870	8.525	3.8	10.4*
mm3	17236	19.320	2850 + 2390	8.587	3.3	11.3*
bfly	25640	20.245	4560 + 3424	8.821	3.2	2.3
Geometric Mean:					3.7	4.4

System performance vs FPU performance (bfly)



Summary

- uPs and C-FPGAs have their strengths and weaknesses for floating point applications
- FP-FPGAs offer a new direction for research
 - Performance evaluation (VEB)
 - Architecture
 - Applications

Questions

- What is the best architecture for an FP-FPGA?
 - Number and functionality of FPUs
 - FPGA: interconnect, memory subsystem, LC granularity
 - Runtime reconfiguration
 - Config bits should be shared among LCs c.f. fine grained FPGA
 - Flash-based configurations, download entire program once to FPGA
- Will it be fast?
 - May lose out for scalar operations
- FP-FPGAs or uPs with reconfigurable FP datapaths?

Spare slides

Area Model

- Estimates of logic cell area including configuration bit, buffer and interconnect overheads.
- A based on estimate that 70% of the total die area for logic cells, the other area being for pads, block memories, multipliers etc.
- Normalised to feature size

Device	LCs/CLB L	Area/CLB A (μm^2)	Feature Size f (μm)	Normalised LC area ($N = A/Lf^2$)
Apex 20K400E [8]	10	63161	0.18	195,000
Virtex E [8]	4	35462	0.18	267,000
Virtex II 3000 [9]	8	$71,429 \times 0.7$	0.12	434,000
Virtex II 1000 [10]	8	$72,782 \times 0.7$	0.12	442,000