CEG 5010: Reconfigurable Computing
The Density Advantage of FPGAs

Philip Leong

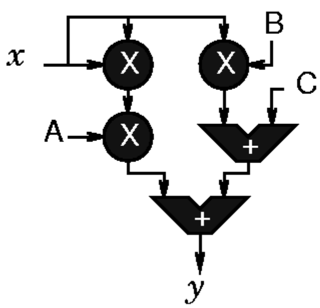
"The world is moving so fast these days that the man who says it can't be done is generally interrupted by someone doing it." - Harry Emerson Fosdick

FPGA vs uP

- Primitive element: LUTs implement a single instruction
  - Implement tasks by spatially composing primitive operations
  - LUT performs same operation every cycle
- uP: temporally compose operations by sequencing them in time
- FPGAs have an area advantage at the cost of restricting the size of the computation

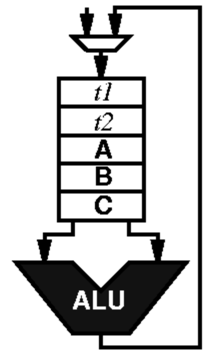
Spatial vs. Temporal Computing

Spatial



Temporal

$$t1 \leftarrow x$$
$$t2 \leftarrow \mathbf{A} \times t1$$
$$t2 \leftarrow t2 + \mathbf{B}$$
$$t2 \leftarrow \mathbf{t2} \times t1$$
$$y \leftarrow \mathbf{t2} + \mathbf{C}$$



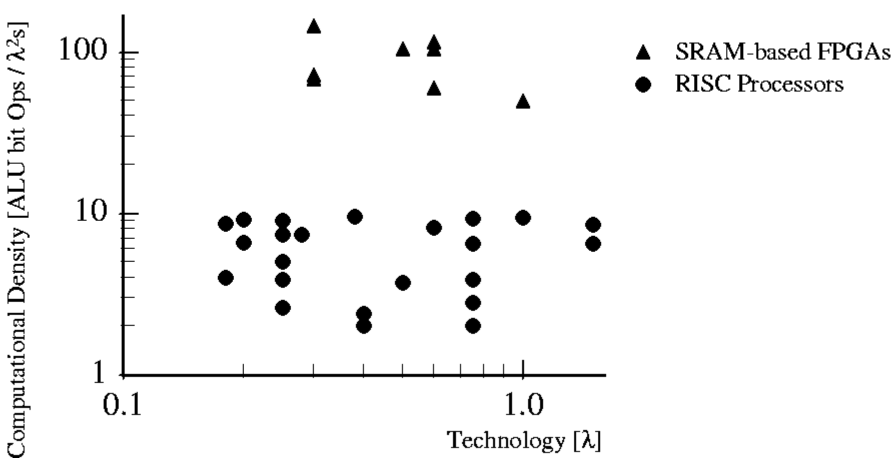
Computational Density vs uP

- FPGAs can complete more work per unit time
  - Less instruction overhead leads to more active computations in same area. More parallelism
- FPGA operations bit level
  - uP word level and often wastes computational capacity when operating on narrow width data

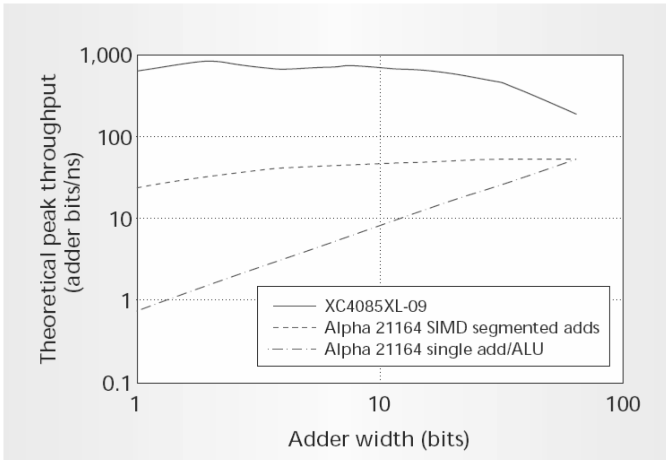
Examples

- Alpha 21164 vs Xilinx XC4085XL-09
- Both 0.35um
- 21164 has two 64-bit ALUs @433MHz
  - 128 bit operations per 2.3 ns = 55.7 bit ops/ns
- XC4085 has 6,272 LCs @ 4.6ns (peak)
  - 3,136 bit ops per 4.6 ns = 62 bit ops/ns
- Difficult to achieve FPGA's peak performance

Density Comparison



Max adder throughput



Multiplier Efficiency

Table 2. Area-time and ratio comparisons of various multipliers. Ratios are shown in parentheses.

	Area-time (λ^2s) and ratio to custom device			
Device style	16 × 16	16 × 16-bit constant	8 × 8	8 × 8-bit constant
Custom	0.104 (1)	0.104 (1)	0.104 (1)	0.104 (1)
FPGA	13.2 (130)	4.2 (41)	3.3 (32)	0.69 (6.6)
DSP	17.5 (170)	17.5 (170)	17.5 (170)	17.5 (170)
Processor	363 (3,500)	57.8 (560)	198 (1,900)	33 (320)

- Factors which affect density: overgenerality and lack of use

FIR filter efficiency

Table 3. FIR survey: 8-bit samples, 8-bit coefficients (LE: logic element).

Architecture	Design	Feature size (μm)	Area	Time	Area-time/tap (λ^2s)
32-bit RISC	Yetter et al.[10] Magenheimer et al.[11]	0.75	125 million λ^2	66 ns/cycle × 6+ cycles/tap	50
16-bit DSP	Kaneko et al.[9]	0.65	350 million λ^2	50 ns/tap	17.5
32-bit RISC/DSP	Nadehara et al.[13]	0.25	1.2 billion λ^2	40 ns/tap	46
64-bit RISC	Gronowski et al.[4]	0.18	6.8 billion λ^2	2.3 ns/tap	16
XC4000	Newgard[14]	0.60	240 CLBs × 1.25 million λ^2/CLB	14.3 ns/8 taps	0.54
Altera 8000	Altera[15]	0.30	30 LEs × 0.92 million λ^2/LE	10 ns/tap	0.28
Full custom	Ruetz[16]	0.75	400 million λ^2	45 ns/64 taps	0.28
	Golla et al.[17]	0.60	140 million λ^2	33 ns/16 taps	0.28
	Reuver and Klar[18]	0.75	82 million λ^2	50 ns/10 taps	0.41
Full custom (fixed coefficient)	Laskowski and Samueli[19]	0.60	114 million λ^2	6.7 ns/43 taps*	0.018

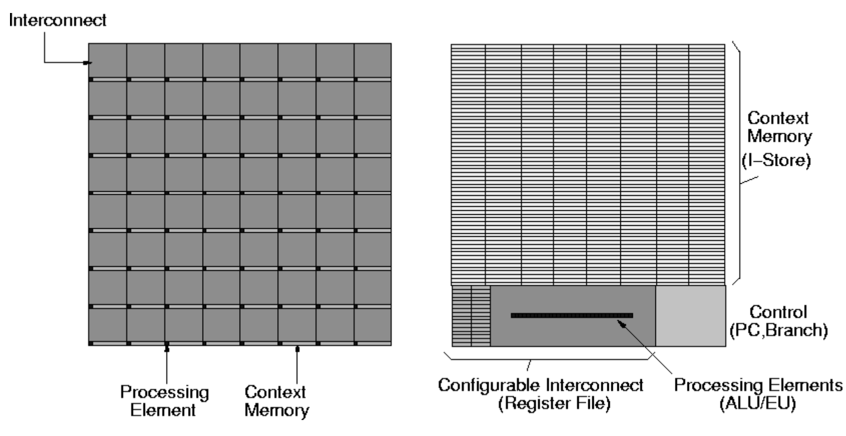
*16-bit samples

FPGAs and uPs

Spatial/Configurable Benefits

- 10x raw density advantage over processors
- potential for fine-grained (bit-level) control --- can offer another order of magnitude benefit

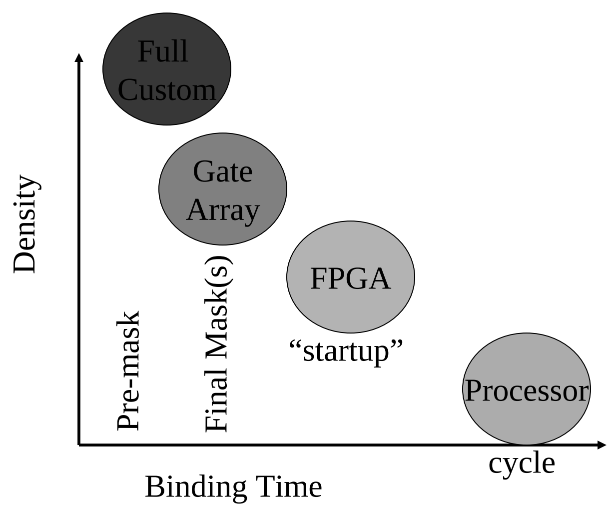
Processor vs. FPGA Area



Configurable Drawbacks

- Each compute/interconnect resource dedicated to single function
- Must dedicate resources for every computational subtask
- Infrequently needed portions of a computation sit idle --> inefficient use of resources

Density vs. Binding Time



Efficiency vs data width

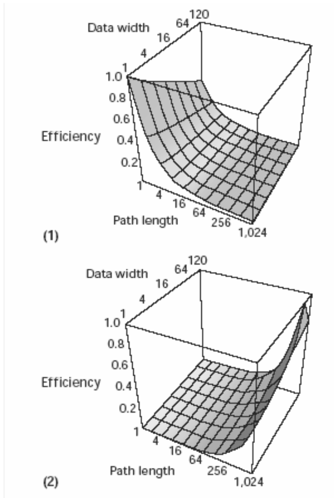


Figure C. Design efficiency at varying application data widths and path lengths of (1) an FPGA and (2) a processor.

Where CC interesting?

- Regular applications -- need same operation repeatedly
- High concurrency -- large number of operations can occur simultaneously
- Fine-grained data -- small operand data widths

Conclusion

- Post-fabrication programmable computing space >> processor arch.
- Challenges
  - Better tools to make designing RC systems easier
  - Novel computer architectures which use this advantage
  - Using late binding advantageously
  - Hybrid archiectures e.g. FPGA+uP (Triscent, Altera, Xilinx)
  - Novel reconfigurable computing architectures (e.g. dynamic configuration)

References

- A. DeHon, “The Density Advantage of Configurable Computing”, IEEE Computer April 2000, pp. 41-49.