FPGAs for IEEE Floating Point

FPGA combine many of the features needed for high performance in numerical applications:
· High parallelism – hundreds of multipliers
· Massive memory bandwidth – hundreds on on-chip RAMs
· Moderate power consumption – 50% of Xeon, 25% of nVidia

New Altera compilation technology makes more floating point capability available

High non-chip memory bandwidth keeps operands supplied.

Stratix ® EP3SE260 Memory Bandwidth

Memory	Number	Width	Ports	Bit/clk
M144	48	72	2	6,912
M9K	864	36	2	74,880
MLAB	5100	20	2	204,000

Technology	GFlops	Precision	Power (W)	GFlops/W
Altera 3SE260	82.0	Single	30	2.7
Xeon Quad 54XX	70.0	Single	70	1.0
nVidia C870	70.0	Single	150	0.5
Xeon Quad 54XX	43.1	Double	70	0.6
Altera 3SE360	44.2	Double	30	1.3

Fused data paths reduce gate count and latency

Typical floating point bloc:
- Prenormalize operands
- Calculate
- Renormalize result,

Redundant normalization wastes gates and cycles.

Instead, consider each operation in the context of adjacent operations.
Guard bits eliminate need to post-normalize at every step.
Instead, worst-case analysis picks specific points for normalization.
Choose context-specific versions of each operator to minimize logic.

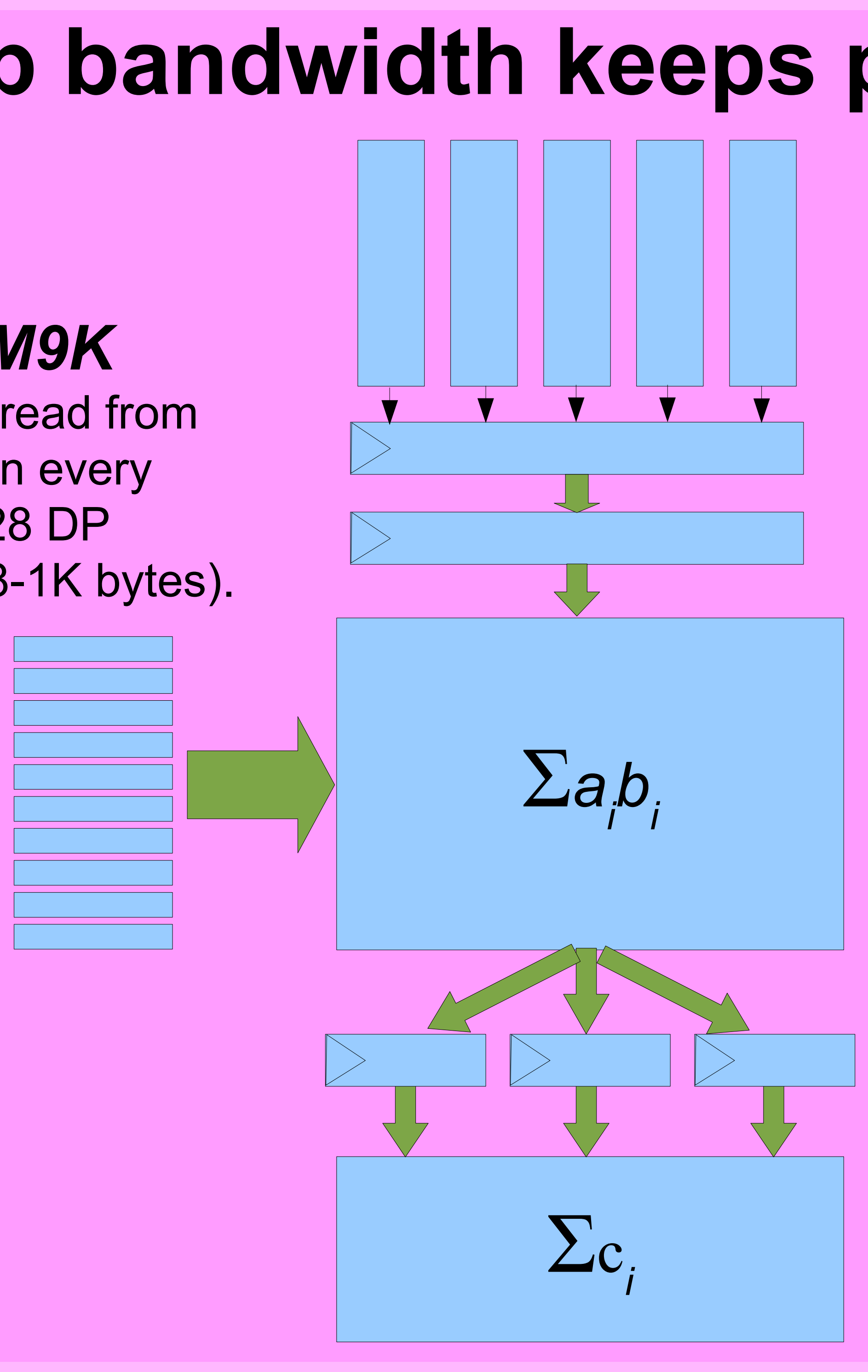
```
dp00 = ((xx00*cc00 + xx01*cc01) + (xx02*cc02 + xx03*cc03)) +
       ((xx04*cc04 + xx05*cc05) + (x06*cc06 + xx07*cc07));
dp01 = ((xx08*cc08 + xx09*cc09) + (xx0a*cc0a + xx0b*cc0b)) +
       ((xx0c*cc0c + xx0d*cc0d) + (xx0e*cc0e + xx0f*cc0f));
dp02 = ((xx10*cc10 + xx11*cc11) + (xx12*cc12 + xx13*cc13)) +
       ((xx14*cc14 + xx15*cc15) + (xx16*cc16 + xx17*cc17));
dp03 = ((xx18*cc18 + xx19*cc19) + (xx1a*cc1a + xx1b*cc1b)) +
       ((xx1c*cc1c + xx1d*cc1d) + (xx1e*cc1e + xx1f*cc1f));
result = ((dp00+ p01) + (dp02+dp03));
```

Result compared to naïve block assembly:
Gate count reductions to 40%
Latency reductions to 40%

"Floating point data path synthesis for FPGAs,", Martin Langhammer, Proc. FPL 2008

On-chip bandwidth keeps pipeline full

B array: M9K
New column read from memories on every cycle: 16-128 DP values, (128-1K bytes).



A array: M144K
Multi-cycle read of row. Row is re-used for multiple columns

Fused data path
Dot product for part of matrix multiply

Partial sum buffers

Fused data path
Final sum

Results

Floating Point performance:
47.46 Gflops IEEE double precision, *until throttled by system bus!*

No hard floating point cores
Achievable with standard parts

Sustained throughput
Major memories double-buffered