ARM FPUs: Low Latency is Low Energy

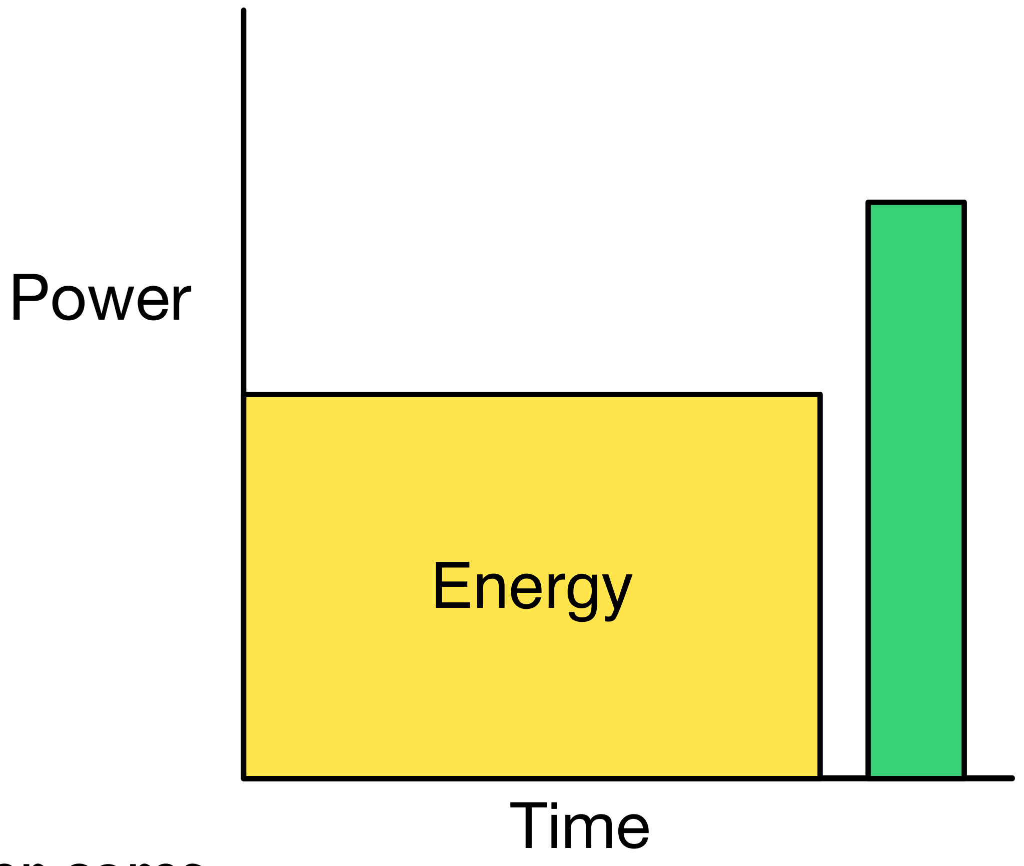
David Lutz

Every computer has a power budget

device	simple phone	smartphone	tablet	laptop	supercomputer
total power budget	3W	5W	15W	35W	20 megawatts
screen size	3"	4-5"	10"	13"	

- Power limited by heat generated
- Performance increases over time, but power budget does not
- Active research area: how to get more performance within a power budget

Low Latency is Low Energy



- Energy = Power * Time
- Datapaths consume little power on out-of-order cores
  - Current ARM FPUs consume about 7% of "big" core power running DAXPY
- Decreasing latency can decrease time
- Energy savings is not just datapath energy

ARM

Typical 5-cycle FMA

- all 3 operands needed at the beginning of the operation
- sum of 4 products: $s = a*x + b*y + c*z + d*w$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
fmul s,a,x	M	M	M	M	M															
fma s,b,y						F	F	F	F	F										
fma s,c,z											F	F	F	F	F					
fma s,d,w																F	F	F	F	F

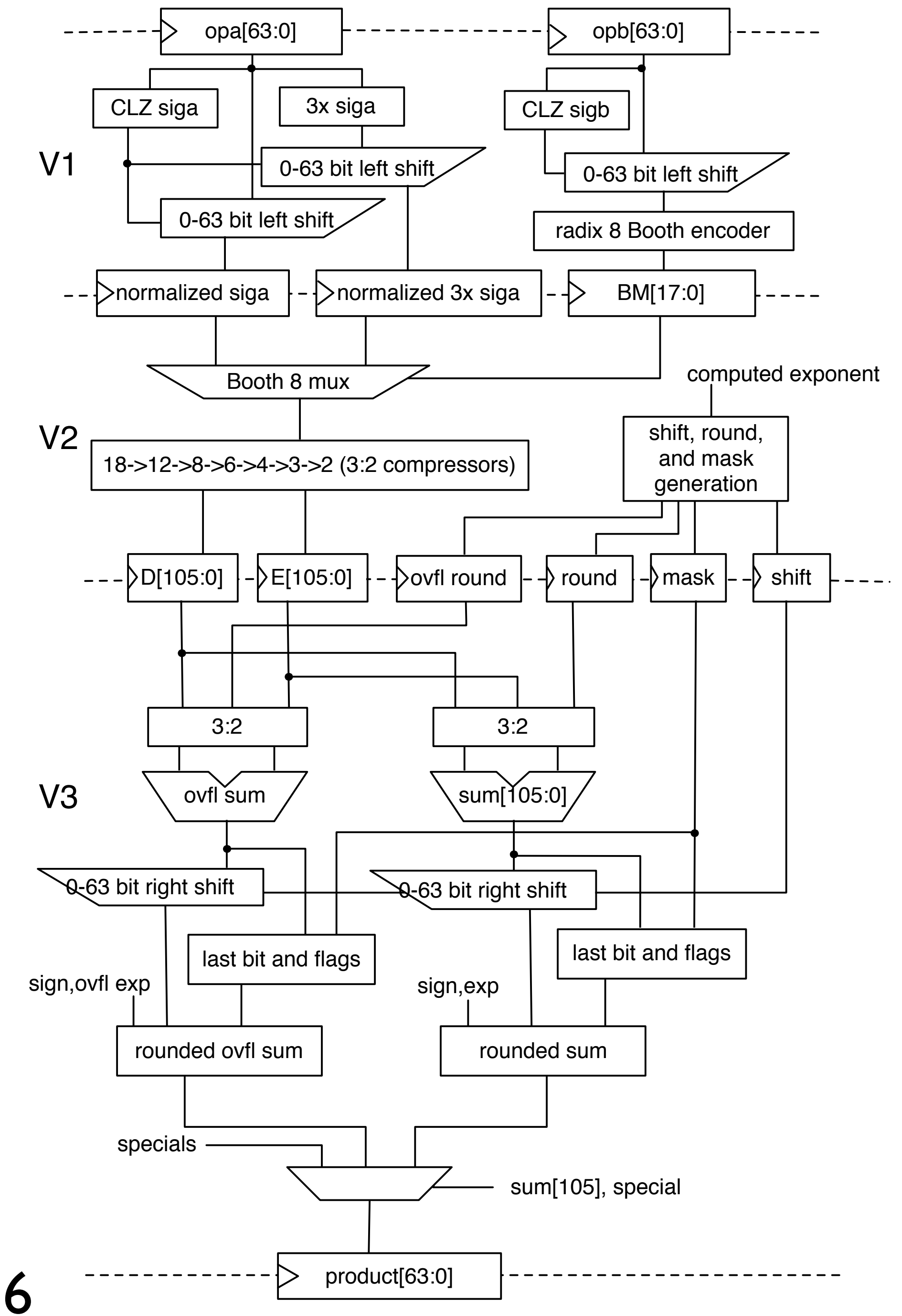
ARM 6-cycle FMA with separate multiply and add

- 3-cycle multiply followed by 3-cycle add
- Note that a single FMA is slower
- sum of 4 products: s = a*x + b*y + c*z + d*w

	1	2	3	4	5	6	7	8	9	10	11	12	13
fmul s,a,x	M1	M2	M3										
fma s,b,y		M1	M2	M3	A1	A2	A3						
fma s,c,z					M1	M2	M3	A1	A2	A3			
fma s,d,w								M1	M2	M3	A1	A2	A3

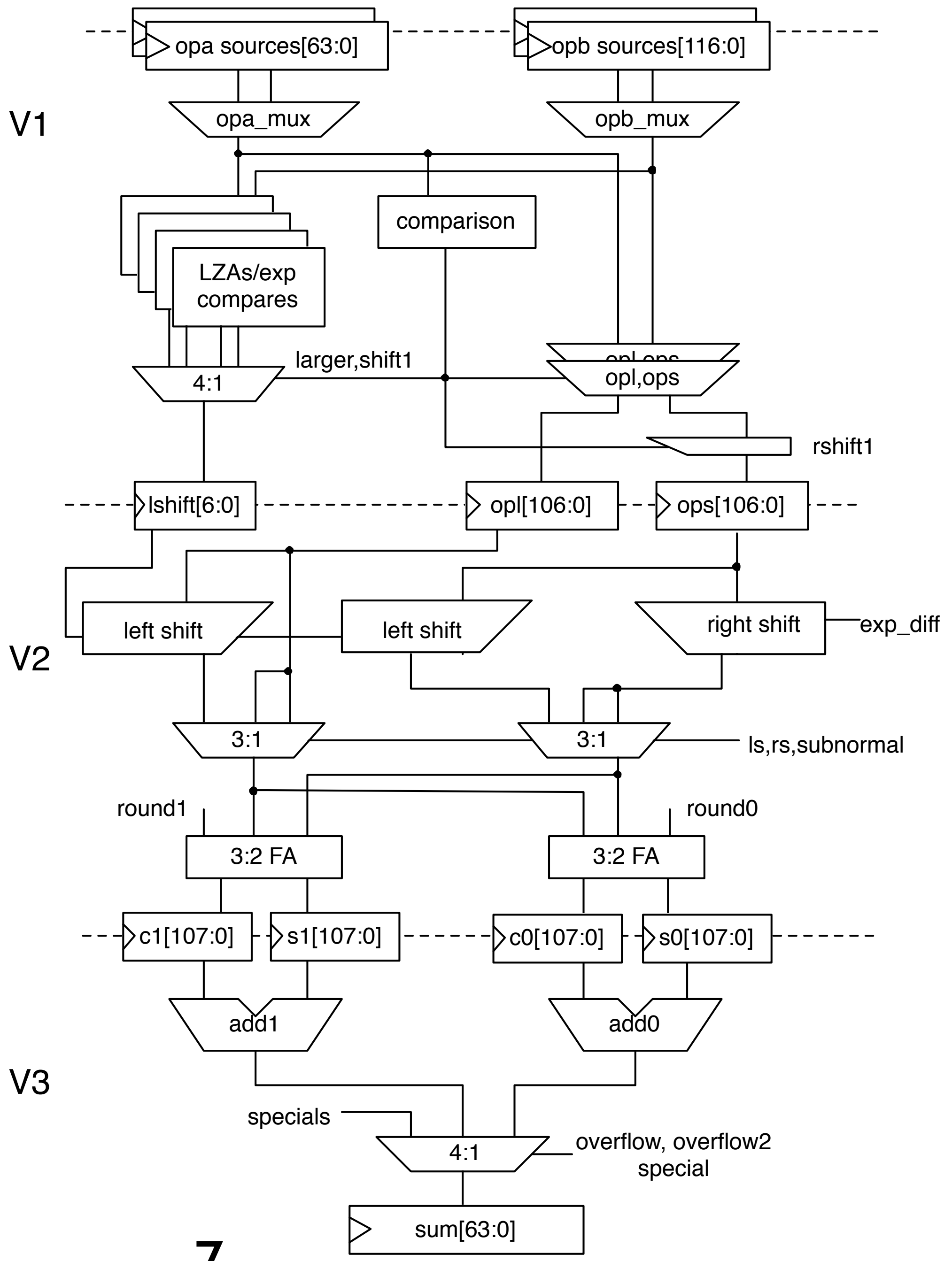
3-cycle multiplier

- V1
  - normalization
  - Booth encoding
- V2
  - Booth mux
  - 18:2 reduction
  - compute shift,round,mask
- V3
  - add and round (2)
  - subnormal shift
  - select



3-cycle adder

- V1
  - compare/swap
  - 4xLZA
  - compute exponent
  - compute lshift, rshift
- V2
  - Left and right shift
  - select
  - 3:2 for rounding
- V3
  - add and round
  - select



Faster FPU = higher performance *and* lower energy

- Suppose lower latency FPU is 15% faster than higher latency FPU
- Takes 1/1.15 = .87 of the time to complete SpecFP

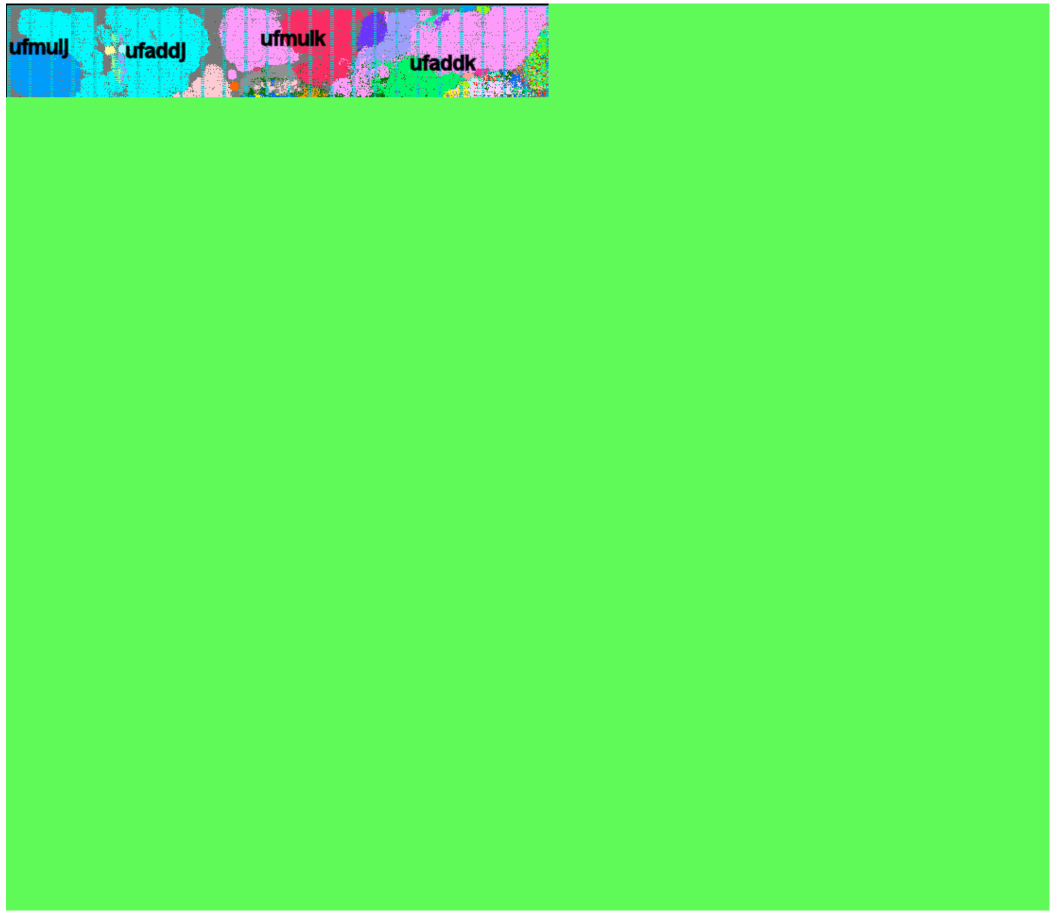
	time	FP power	non-FP power	energy = time * power
Slower FPU	1	7	93	1.0 * (7+93) = 100
Faster FPU	0.87	p	93	.87 * (p+93) = .87p + 80.9

- New scheme lower energy if 100 > .87p + 80.9
  - if p < 22
  - if p < 3 times slower FPU power

ARM

Faster FPU can lead to lower area

- Fewer (flip)flops vs. more logic
- Where is the area going?



Strategy for out-of-order cores

- Do the execution as quickly as possible to save energy
- Be suspicious of slower execution, e.g.
    - double pumped multipliers
    - slower dividers
- Execution units are where you want to spend power

Conclusions

- Low execution latency has an outsized effect on performance
- Low latency can improve area
- Low latency is low energy