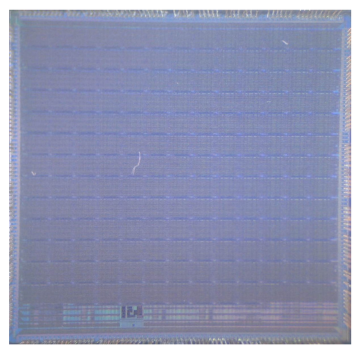
Massively Parallel Processor Array for Mid-/Back-end Ultrasound Signal Processing

Energy Efficient Parallelism

Dean Truong
Bevan Baas
VLSI Computation Lab
University of California, Davis

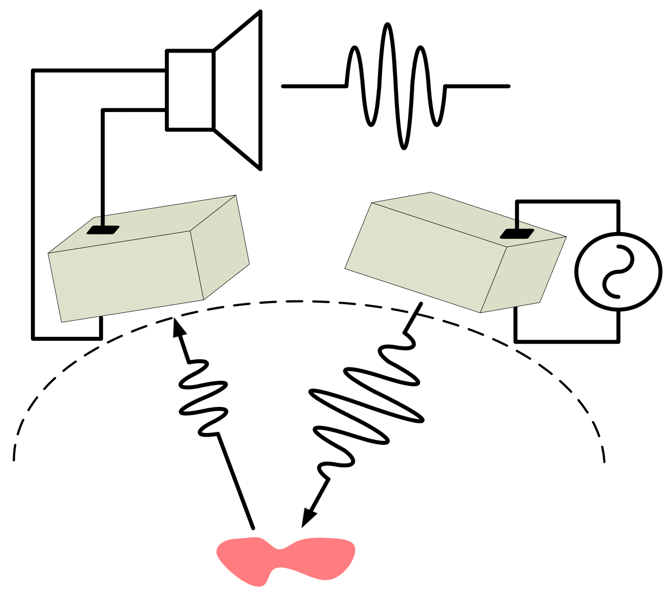
Many-core Energy Efficiency

- Recent interest in portable ultrasound machines
  - Extended battery life desired
  - Performance compromise must be minimal
- Energy efficiency through parallelism
  - Ultrasound processing similar to DSP in radar, sonar, etc.
  - DSP applications tend to exhibit task level parallelism
- Many-core chips can take advantage of parallelism
  - Thousands of cores in future nanometer CMOS technologies



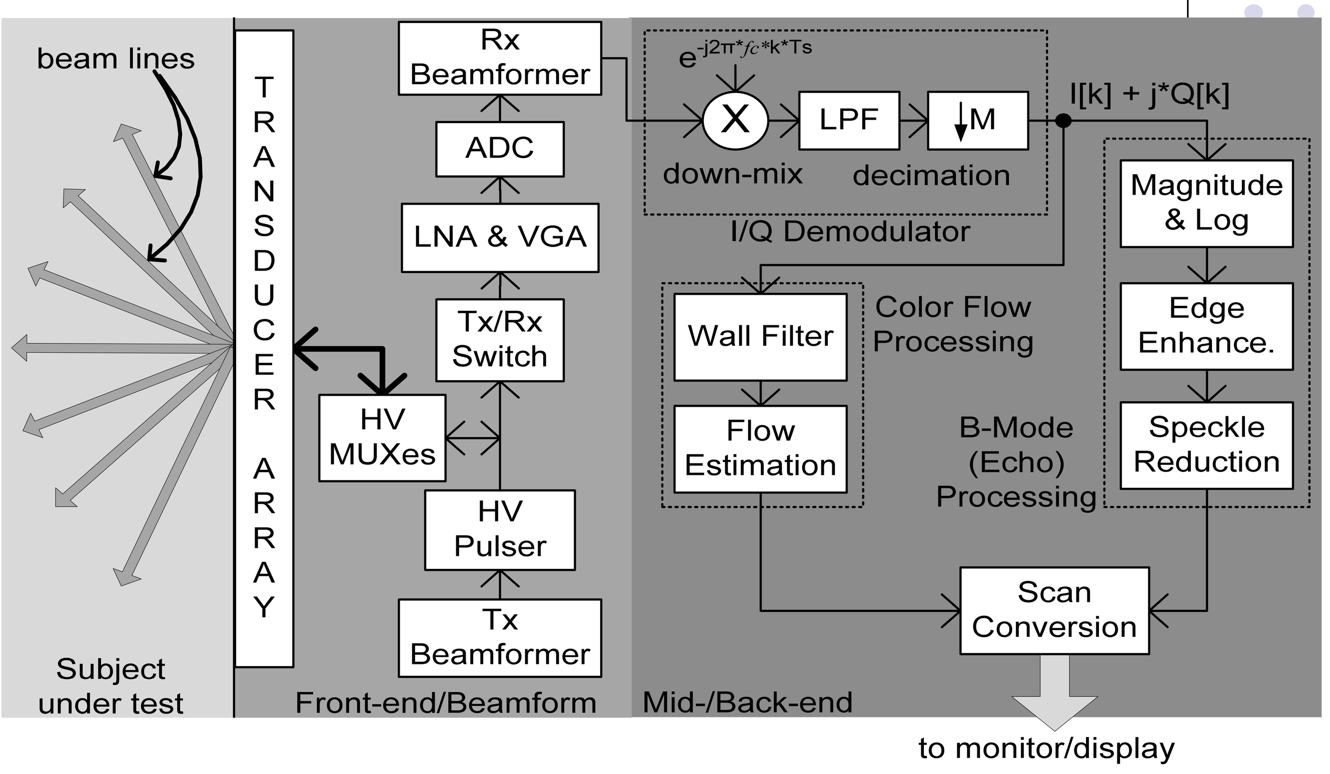
65 nm 167-core Asynchronous Array of simple Processors

Ultrasound Imaging Modality



- Ultrasound uses high frequency sound (2-10 MHz)
- The sound is able to penetrate the skin and reflect off of objects of interest
  - The intensity of reflection varies with an object's acoustic impedance
- Piezoelectric ceramic transducers are used to send and receive ultrasonic pulses
  - Echoes caused by objects along the path of the initial pulse are processed into images

Basic Ultrasound System

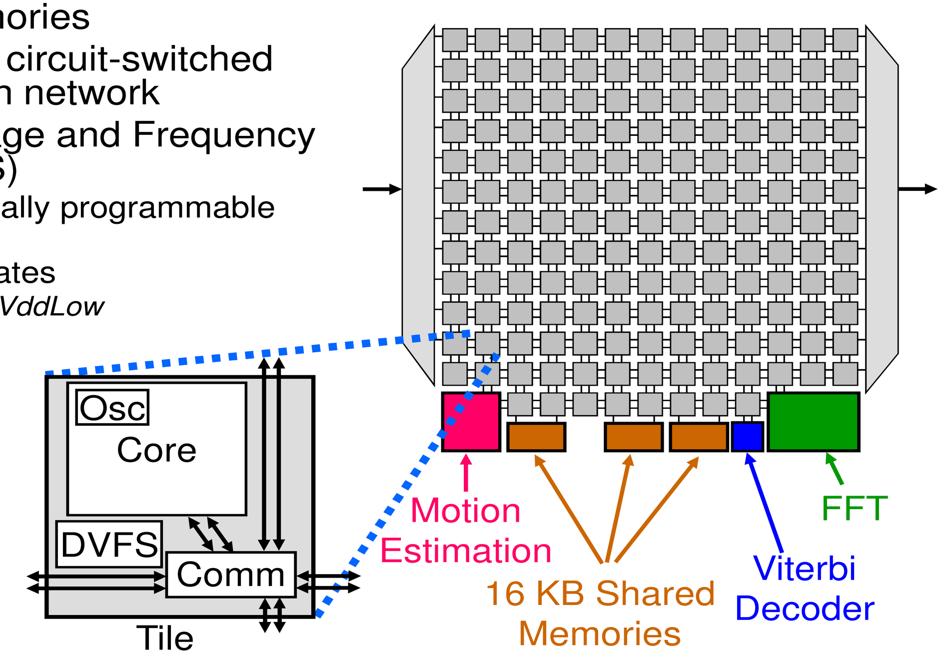


Many-core DSP Architecture

- 164 Simple DSP processors
- 3 Dedicated-purpose processors
- 3 Shared memories
- Long-distance circuit-switched communication network
- Dynamic Voltage and Frequency Scaling (DVFS)
  - Per-core digitally programmable ring oscillator
  - Two power gates
    - *VddHigh & VddLow*

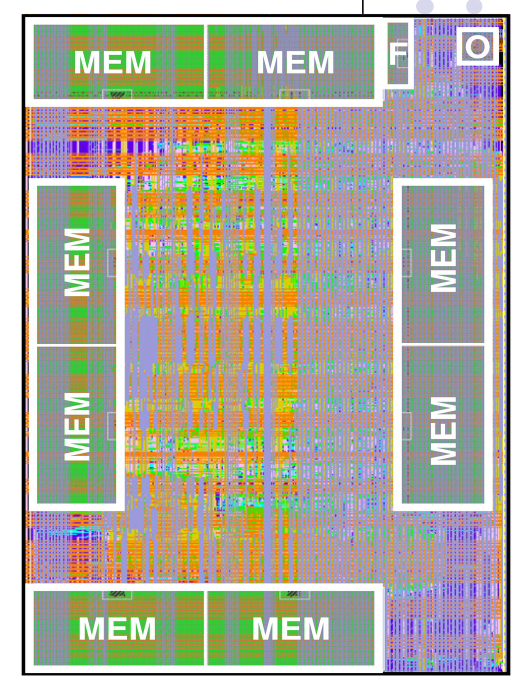
Each core only contains a 128 word DMem and a 128 word 16-bit IMem.

FIFOs used for interprocessor communication

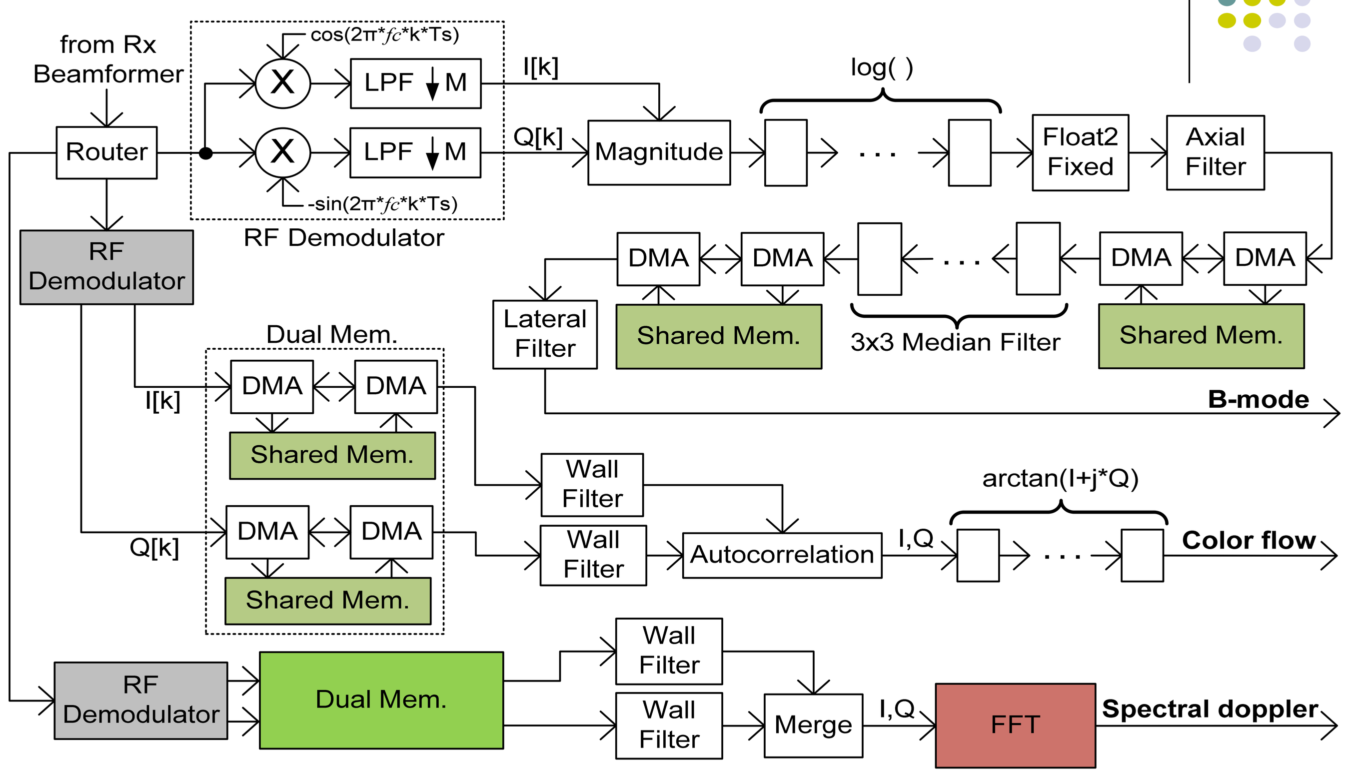


FFT Processor

- Runtime configurable
  - 4- to 4096-point FFT/IFFTs
- 32-bit fixed-point complex data
- High SQNR across all modes:
  - ~80 dB for 64-point
  - ~74 dB for 1024-point
- 866 MHz, 34.97 mW @ 1.3 V
- 67 ns to compute a 64-point FFT
  - Over 950 Msamples per second
- 1.5 µs to compute a 1024-point FFT
  - Over 680 Msamples per second
- In our ultrasound implementation the FFT only needs to run at 4 MHz
  - Up to 4 Msamples per second



Mid-/Back-end Implementation



Task Parallelization Example

Algorithm 1 CORDIC Arctangent

Require: Get initial values of I and Q

$\theta = 0$
for $n = 0$ to $N-1$ **do**
  $I_{shifted} = I >> n$
  $Q_{shifted} = Q >> n$
  if $Q < 0$ **then**
    $I_{new} = I - Q_{shifted}$
    $Q_{new} = Q + I_{shifted}$
    $\theta_{new} = \theta - \arctan(2^{-n})$
  else
    $I_{new} = I + Q_{shifted}$
    $Q_{new} = Q - I_{shifted}$
    $\theta_{new} = \theta + \arctan(2^{-n})$
  end if
  $I = I_{new}$
  $Q = Q_{new}$
  $\theta = \theta_{new}$
end for

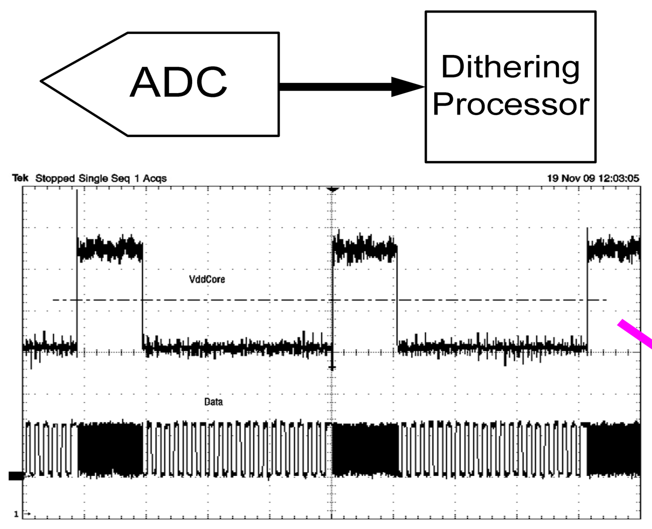
- CORDIC algorithms are easily parallelized into a pipeline of tasks
  - i.e. one iteration of a loop is transformed into one task
  - This is identical to loop unrolling
- We can increase the cycle throughput by *N* and thus be able to run the processors at lower frequencies
- Latency is longer, but frequencies can also be increased to decrease processing time per processor
- For CORDIC atan, three values are updated between iterations (tasks)—these values are passed between processors

Voltage Dithering

- Voltage dithering takes quantized voltages and allows us to effectively interpolate (generate) other voltage levels
  - Energy is dependent only on voltage, so energy efficiency occurs at maximum frequencies
    - **CMOS Energy = αCV^2**
- Voltage selection is treated as a duty cycle
  - Define the duration we stay at *VddHigh* and *VddLow*
    - Percentages: P_H and P_L, respectively
  - Energy ~ $P_L \times (V_L^2) + P_H \times (V_H^2)$

Voltage Dithering (cont.)

- Workload ~ frequency, f_D = desired workload
  - At *VddHigh*, maximum frequency = f_H
  - At *VddLow*, maximum frequency = f_L
- Duty Cycle: $P_L = (f_D - f_H)/(f_L - f_H)$, $P_H = 1 - P_L$



Worst case:
Vdd = 1.3 V, f_H = 1.2 GHz, 46.6 mW

Ideal case:
Vdd = 0.9 V, f_D = 550 MHz, 12.0 mW

Dithering:
VddHigh = 1.3 V, f_H = 1.2 GHz
VddLow = 0.8 V, f_L = 300 MHz
15.6 mW

Ideal: P_L = 72.22%, P_H = 27.78%

Measured: P_L = 74.4%, P_H = 25.6%

Results

- Assume: 10 MHz carrier frequency, decimation factor of 4, and 80 Msamples/sec ADC, with pulse repetition frequency of 19.25 kHz
- 512 B-mode beam lines, 192 color flow beam lines, ensemble size of 8, ~1024 samples per beam line
- Requires: 77 DSPs, 6 shared memories, and one FFT processor
- Cycles per FIFO read (*CPR*) and cycles per FIFO write (*CPW*) used to determine the required operating frequencies for each processor
- Optimal Vdd & frequency average power results approximates the performance of voltage dithering

Processor(s) or Memory	CPR (cycles)	CPW (cycles)	IMEM (instr.)	DMEM (words)	Optimal Frequency (MHz)	Optimal Vdd (Volts)	× (units)	Average Power (mW)
B-mode								
Mixer	4	4	5	5	320	0.8	2	12.36
LPF/Decimate	4	16	22	16	320	0.8	2	12.36
Magnitude	5	5	6	1	100	0.7	1	1.54
Log (PreP1)	8	8	17	3	160	0.7	1	2.47
Log (PreP2)	10.5	5.25	21	3	210	0.75	1	3.63
Log (Core 1)	4.5	3.6	18	5	180	0.7	1	3.12
Log (Core 2 to 13)	3.6	3.6	18	5	180	0.7	12	37.38
Log (Core 14)	3	3.75	15	3	150	0.7	1	2.31
Log (Core 15)	3	4	11	1	120	0.7	1	1.85
Log (PostP)	6	12	32	4	180	0.7	1	3.12
float2fixed	3.5	7	8	3	70	0.7	1	0.81
Axial Filter	31	31	45	16	310	0.8	1	4.02
DMA Write (Med)	2.16	2.16	104	15	21.6	0.67	1	0.23
Memory (Med)	1	1	N/A	N/A	10	1.3	1	0.03
DMA Read (Med)	2.45	2.45	55	10	24.5	0.67	1	0.26
Median 0 to 3	2.56	2.56	7	1	25.6	0.67	4	1.10
Median 4	2.56	2.875	7	1	25.6	0.67	1	0.28
Median 5	2.875	3.833	7	1	25.6	0.67	1	0.28
Median 6	3.167	4.75	7	1	21.2	0.67	1	0.23
Median 7	3.75	7.5	7	1	16.8	0.67	1	0.18
Median 8	5.5	11	6	1	12.3	0.67	1	0.13
DMA Write (Lat)	2.75	2.75	69	10	3.1	0.67	1	0.03
Memory (Lat)	1	1	N/A	N/A	1.2	1.3	1	0.01
DMA Read (Lat)	2.45	2.45	35	10	2.8	0.67	1	0.03
Lateral Filter	1.6	8	9	7	1.9	0.67	1	0.02
Subtotal							41	87.79

Processor(s) or Memory	CPR (cycles)	CPW (cycles)	IMEM (instr.)	DMEM (words)	Optimal Frequency (MHz)	Optimal Vdd (Volts)	× (units)	Average Power (mW)
Color Flow								
Mixer	4	4	5	5	320	0.8	2	12.36
LPF/Decimate	4	16	22	16	320	0.8	2	12.36
DMA Write	2.75	2.75	69	10	55	0.7	2	1.70
Memory	1	1	N/A	N/A	20	1.3	2	0.14
DMA Read	2.41	2.41	35	10	48.2	0.7	2	1.49
Wall Filter	1.6	8	9	7	32	0.67	2	0.92
Autocorrelation	7	7	15	5	28	0.67	1	0.40
Arctan (PreP)	1.5	1	6	0	6	0.67	1	0.09
Arctan (Core)	2.75	2.75	12	5	16.5	0.67	14	3.32
Arctan (PostP)	5	10	8	1	30	0.67	1	0.43
Subtotal							29	33.20
Spectral Doppler								
Mixer	4	4	5	5	320	0.8	2	12.36
LPF/Decimate	4	16	22	16	320	0.8	2	12.36
DMA Write	2.75	2.75	69	10	55	0.7	2	1.70
Memory	1	1	N/A	N/A	20	1.3	2	0.14
DMA Read	2.41	2.41	35	10	48.2	0.7	2	1.49
Wall Filter	1.6	8	9	7	32	0.67	2	0.92
Merge	2	1	3	0	8	0.67	1	0.11
FFT	0.5	0.5	N/A	N/A	4	1.3	1	0.16
Subtotal							14	29.24
Total							**84**	**150.23**

Conclusions

- Mid-/Back-end processing can be done efficiently with a many-core array of simple DSP processors
  - Task level parallelism is ubiquitous in ultrasound signal processing
- Massive fine-grained parallelism is used to increase energy efficiency by lowering required operating frequencies to maintain throughput
  - Power density is also reduced by loading multiple cores with lighter workloads
- DVFS capability and dithering approach to reach the optimal operating point of each processor as determined by the *CPR* and *CPW*—150.23 mW total average power
  - Average energy per frame for B-Mode at 37.6 fps and color flow at 12.5 fps is 2.33 mJ/frame and 2.66 mJ/frame respectively
  - Compare this to a static two voltage operation with *VddHigh* = 0.8 V and *VddLow* = 0.67 V, which results in a total average power of 160.07 mW (6.5% increase)

Acknowledgements

- ST Microelectronics
- NSF Grant 0430090, 0903549 and CAREER award 0546907
- C2S2 Grant 2047.002.014
- Intel
- SRC GRC Grant 1598, 1971 and CSR Grant 1659
- Intellasys
- UC Micro
- SEM
- J.-P. Schoellkopf, K. Torki, S. Dumont, Y.-P. Cheng, R. Krishnamurthy and M. Anders