Viterbi Decoding on the ARM

Philip Chong

UC Berkeley CAD Group

May 4, 1999

ARM Overview

- 32-bit RISC microprocessor
- Five stage pipeline
- Features fast ALU operations (barrel shifter)
- Scalar integer unit, no FPU

Algorithm Tweaking

- Metric computation through table lookup (load = 1 delay slot) faster than using ALU (multiplication = up to 3 delay slots)

- Parity computation (Viterbi code) also through lookup

Reducing Memory Footprint

- Willing to give up some algorithmic efficiency to eliminate cache misses
- Pack 32 bits of traceback into single word; easy retrieval due to barrel shifter
- For 128 level traceback, memory requirements are 512 B metrics + 1024 B traceback + 768 B lookup tables = 2304 B

Quality of Results

- 16 bit fixed point, 128-level traceback gives 1.50e-4 BER, 0.196dB SNR degradation (identical to reference implementation)

- Reduction to 64-level traceback gives 7.00e-4 BER, 0.725dB SNR degradation

- Single-bit encoding gives 1.36e-2 BER, 1.963dB SNR degradation

- Choose to stick with 16 bit, 128-level traceback

Simulation

- Used instruction-level simulator by Prof. Rabaey's group
- Model available for ARM8 core at $3.3V$ (also SA-110)
- Characterized at $125MHz$; will have to extrapolate
- Includes power for cache ($8kB$ unified)

Simulation Results

- Simulated decoding of 4096 bits
- 11.72M instruction cycles
- 52.47mW at 125MHz, 3.3V
- Gives 44kb/s

Extrapolation

- Existing Intel SA-110 at 275 MHz, 2.0 V core
- $P \propto V^2 f$
- Performance shortcoming due to test overhead; loop unroll should fix

Operating Parameters	Power	Speed
$125MHz, 3.3V$	$52.47mW$	$44kb/s$
$275MHz, 3.3V$	$115.43mW$	$96kb/s$
$275MHz, 2.0V$	$42.40mW$	$96kb/s$

Area

- Note: Code size less than 600 B
- ARM8 3.75$mm \times$ 1.65mm in 0.25μm
- $8kB$ SRAM 3.2$mm \times$ 0.4mm
- Total Area 7.47mm^2

Predictability

- Power simulations accurate to 10%

- Relative unknown is frequency scaling

- Figure given here is probably overestimate due to static dissipation

Summary

Clock Speed	$275MHz$
Execution Performance	$96kb/s$
Power Dissipation	$42.40mW$ $5.68mW/mm^2$
Area	$7.47mm^2$ in $0.25\mu m$
Design Effort	4 days

- Portability very high: code is ANSI C; architecture-dependent tweaks may need reworking

Conclusions/Thanks

- One-bit quantization gives opportunities for performance improvements, at a huge cost in QOR

- Hardware parallelism (vector ops) would benefit greatly

- Many thanks to Marlene Wan for providing the power estimation