Hardware-Software Tradeoffs

Prof. Stephen A. Edwards

sedwards@cs.columbia.edu

Columbia University

Spring 2006

Digital Camera

Operations

Image capture

- Exposure control
- Image processing
- JPEG compression

Storage

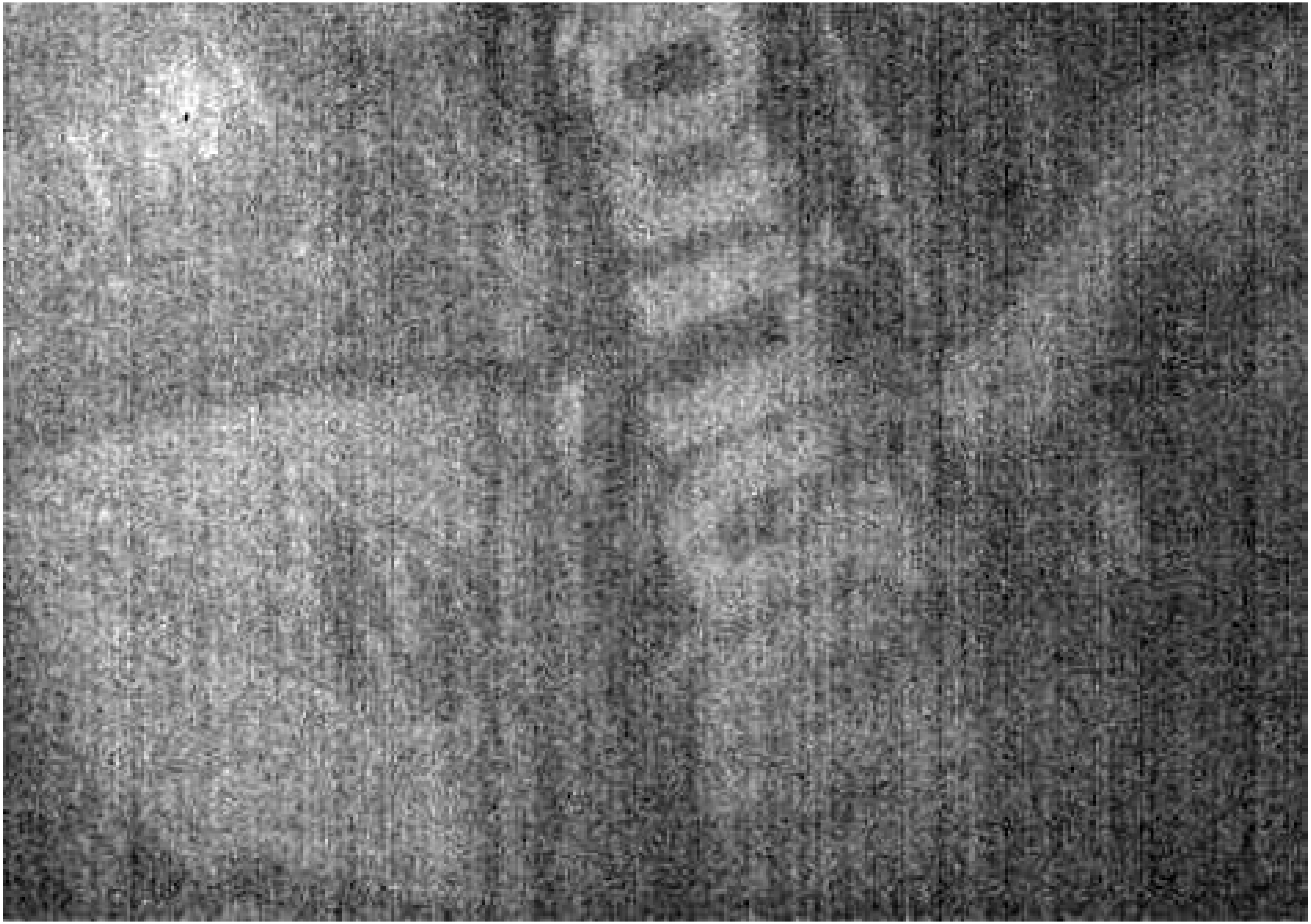
- File system
- CF, SD, SM, or MS interface

Upload

- Serial, USB, or Firewire protocol

Raw Image

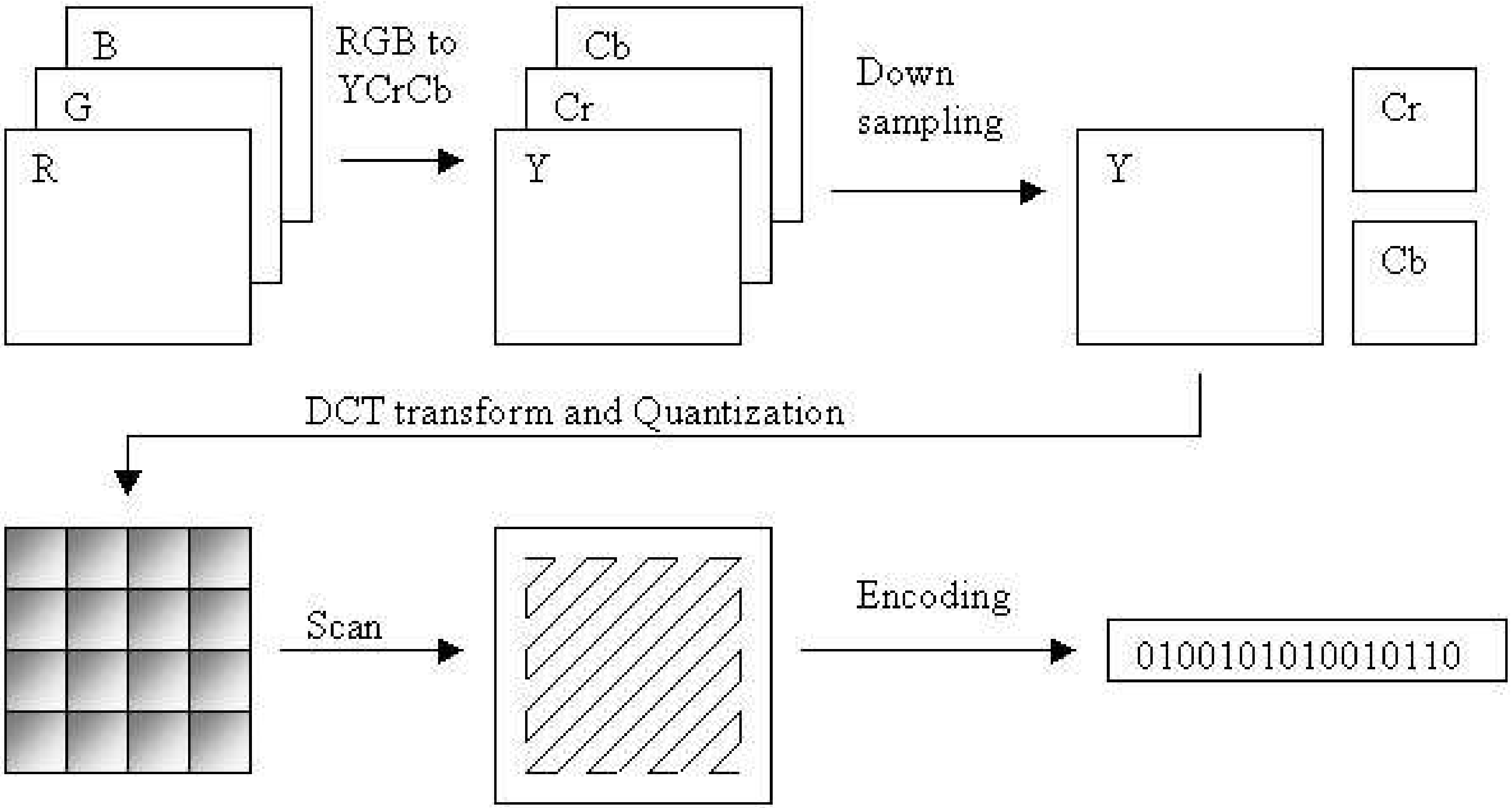
Exposure Correction



Column Bias Correction

Noise Reduction

JPEG Compression



Initial specification

- Stores 50 low-res images
- Can be uploaded to PC
- Retail cost of $100 or less
- Long battery life
- Expected volume of 200 000 if enters market in $<$ 6 months
- 100 000 if in 6–12 months
- Pointless beyond a year

Metrics

- Performance (sec): 1 sec shutter-to-shutter time
- Size (cm^2): chip area
- Energy (Joules): Average consumed per shot
- Resolution (pixels): 64×64
- SNR (dB): 8 bpp, greyscale

Some metrics are *constraints*

Other metrics are to be *optimized*

First step: Build a model

Need some way of estimating design quality

Typical: Build a model in C/C++/SystemC, etc.

Helps to analyze power, computational costs, etc.

Often refined into functional model

Used as golden reference

Design 1: Microcontroller Alone

Intel 8051 microcontroller handles all functions

Cheap: might be $5 per chip in quantity

Low-power: 200 mW

Quick to design: about 3 months

Slow: 12 MHz, 12 cycles per instruction, 1 MIPS

Too slow: CCD zero-bias adjustment takes $\approx$ 100 instructions/pixel

4096 pixels takes half a second; compression even more expensive

Design 2: Zero-bias in hardware

SoC approach: 8051 core + EEPROM + RAM + UART + Zero-bias correction peripheral

Most components standard

Custom hardware increases NRE, time-to-market

Fairly simple custom hardware, though

8051 core modified to add special data instructionsa

Zero-bias hardware: controller, memory access control, ALU, counters

Design 2: Analysis

- Entire system coded in VHDL to verify functionality
- Simulation fast enough to check performance
- Synthesizable: used to obtain area estimates
- Post-synthesis model used for power estimate

Design 2: Analysis Results

9.1 s to process an image: *too slow*

33 mW power consumption

300 mJ energy consumption (9.1s $\times$ 33 mW)

98 000 gates

Simulation shows that processor spends most time performing DCT

Naïve software implementation uses emulated floating-point operations

Better: Use fixed-point arithmetic on the 8051

Design 3: FP DCT + zero-bias HW

Rewrote DCT, otherwise same as Design 2

1.5 s to process an image

33 mW power consumption

50 mJ energy (6$\times$ greater battery life)

90 000 gates because code more compact

Close, but still not fast enough.

DCT biggest time consumer: implement it in hardware

Design 4: HW DCT + zero-bias

Implement complex 8×8 DCT operation in hardware and simulate again

100 ms to process an image

40 mW power consumption

0.4 mJ energy consumption (12× greater than Design 3)

128 000 gates: DCT is a large piece of silicon

Summary of Designs

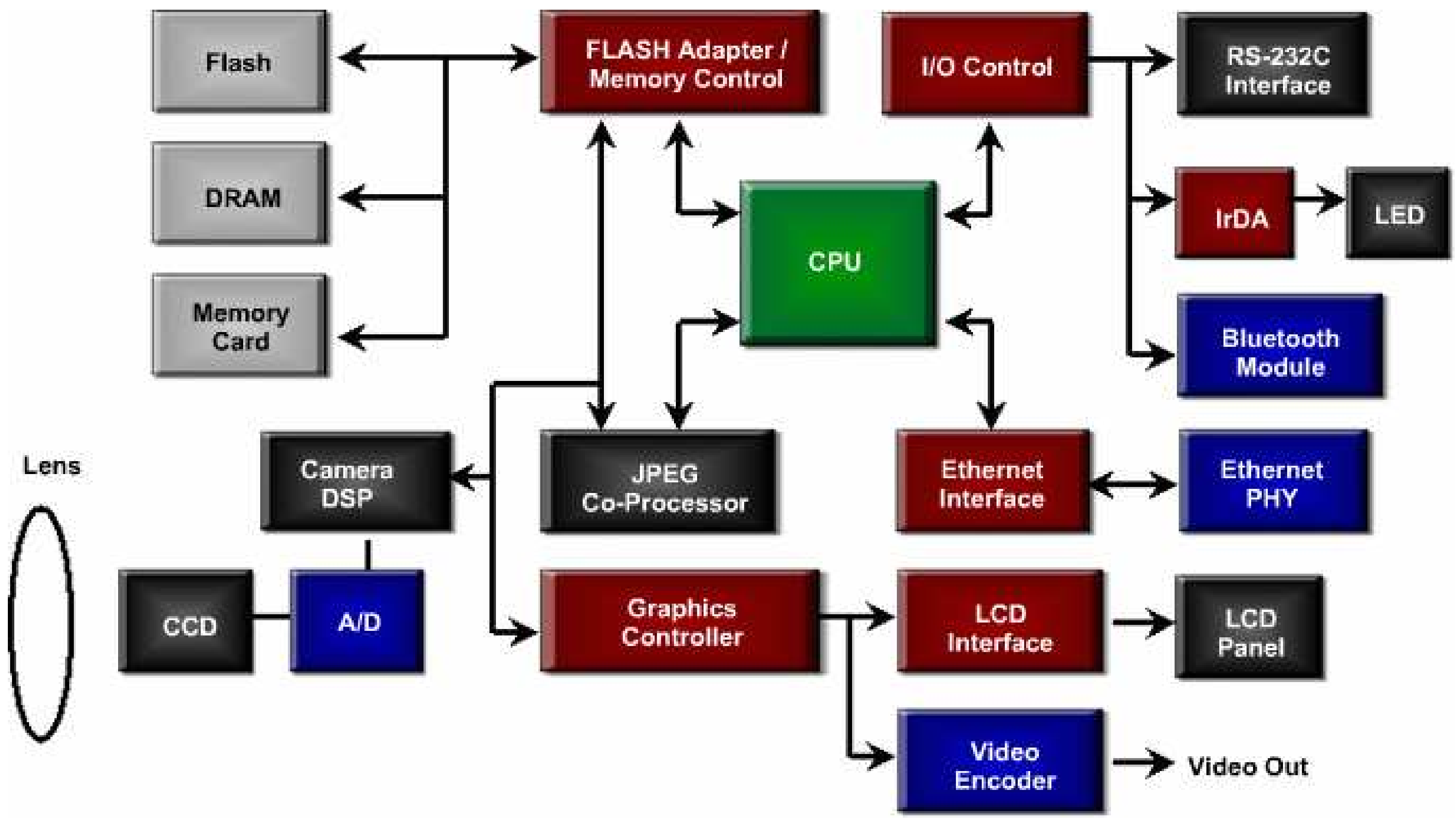
	SW	Zero-Bias	FP DCT	HW DCT
Time (s)	$\gg 1$	9.1	1.5	0.1
Power (mW)		33	33	40
Energy (mJ)		300	50	0.4
Size (kG)		98	90	128

Design 3: FP DCT performance close, cheaper, easier to build

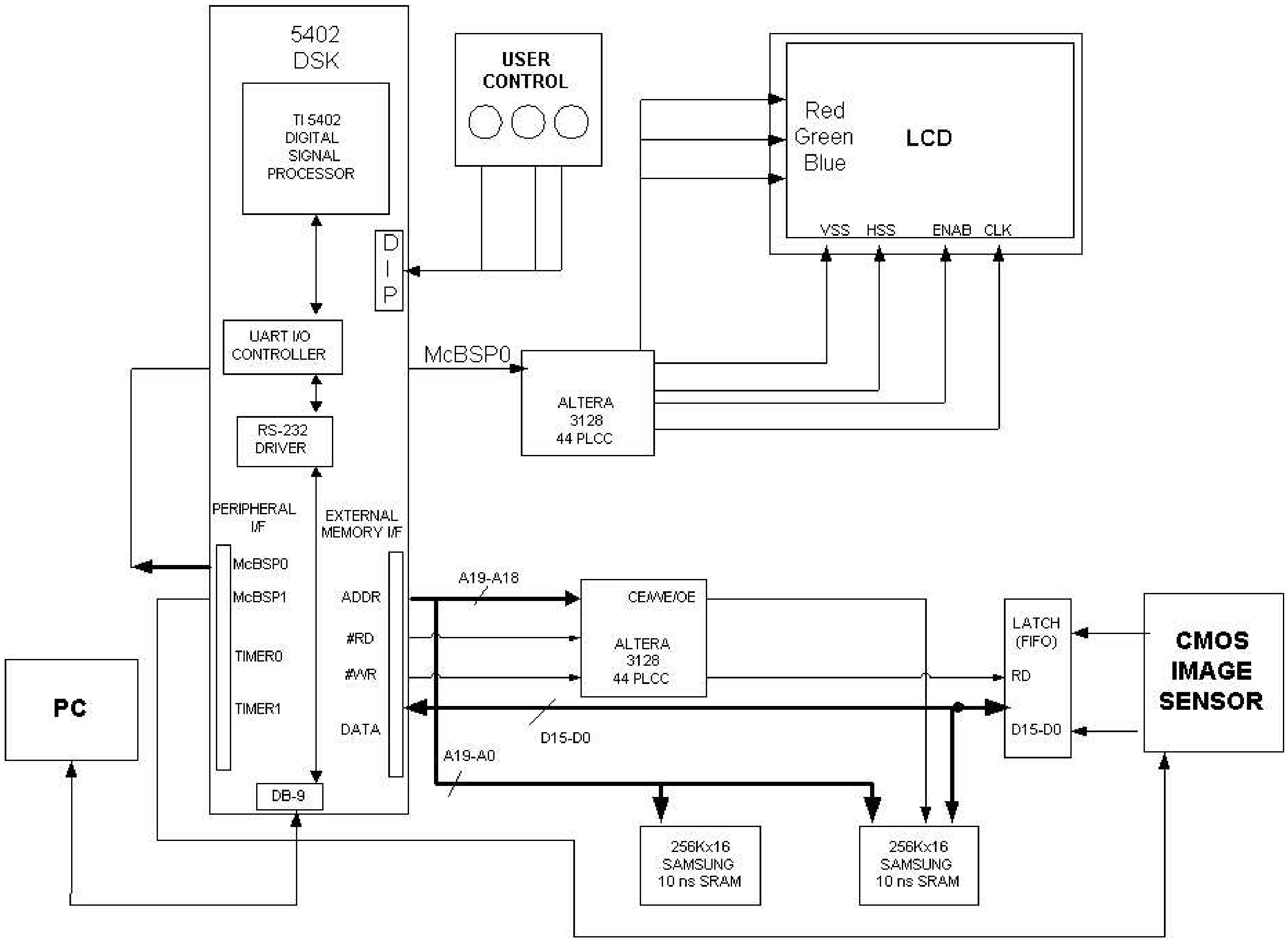
Design 4: HW DCT great performance, energy consumption

Longer to build, may miss market window, may increase IC cost

Example Block Diagram 1



Example Block Diagram 2



Example Block Diagram 3