EEL 4783: Hardware/Software Co-design with FPGAs

Lecture 2: Overview: Digital Camera

Prof. Mingjie Lin

UCF

Stands For Opportunity

Introduction

- Computer-based implementation
  - GPP (General-Purpose Processor)
  - SPP (Special-Purpose Processor)
  - Memory
  - Interfacing
- Designing a simple-yet-complete digital camera
  - Special-purpose IPs
  - Partitioning functionality among different IP cores

What is a Digital Camera?

- Functionality
  - Capturing images
  - Storing digital images (decoding/encoding)
  - Processing digital images (encryption, digitally enhancing, zooming, …)
  - Downloading images to PC
- Implementation
  - Fixed and flexible components
  - Software and hardware components
  - Discrete and integrated components
- Feasibility
  - System-on-Chip
  - High-capacity flash memory

Designer’s Perspective

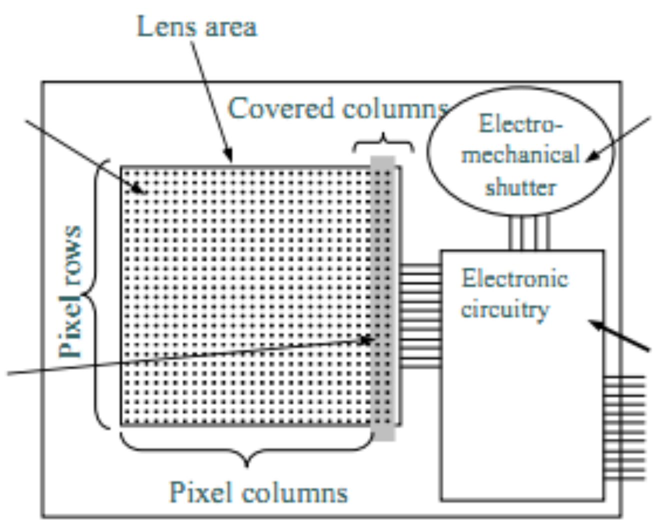
- When shutter pressed:
  - Image captured (CCD: Charge-Coupled Device)
  - Converted digital format
  - Compressed and archived in internal memory
- Transfer images to external platforms
  - Attach camera to a PC
  - Software to grab all images
- Desired properties
  - High-end lens
  - Image capturing speed
  - Resolutions (how many pixels?)
  - Editing capability
  - Supporting format (raw, jpeg, tiff)

Charge-Coupled Device (CCD)

- Special sensor that captures an image
- Light-sensitive silicon solid-state device composed of many cells

When exposed to light, each cell becomes electrically charged. This charge can then be converted to a 8-bit value where 0 represents no exposure while 255 represents very intense exposure of that cell to light.

Some of the columns are covered with a black strip of paint. The light-intensity of these pixels is used for zero-bias adjustments of all the cells



The electromechanical shutter is activated to expose the cells to light for a brief moment.

The electronic circuitry, when commanded, discharges the cells, activates the electromechanical shutter, and then reads the 8-bit charge value of each cell. These values can be clocked out of the CCD by external logic through a standard parallel bus interface.

Zero-Bias Error

- Manufacturing error cause imaging cells to react differently to the same light intensity
- Error same across columns, different across rows
- Reference cells (left-most columns, blocked to detect zero-bias error)

								Covered cells	
136	170	155	140	144	115	112	248	**12**	**14**
145	146	168	123	120	117	119	147	**12**	**10**
144	153	168	117	121	127	118	135	**9**	**9**
176	183	161	111	186	130	132	133	**0**	**0**
144	156	161	133	192	153	138	139	**7**	**7**
122	131	128	147	206	151	131	127	**2**	**0**
121	155	164	185	254	165	138	129	**4**	**4**
173	175	176	183	188	184	117	129	**5**	**5**

Before zero-bias adjustment

Zero-bias adjustment
-13
-11
-9
0
-7
-1
-4
-5

123	157	142	127	131	102	99	235
134	135	157	112	109	106	108	136
135	144	159	108	112	118	109	126
176	183	161	111	186	130	132	133
137	149	154	126	185	146	131	132
121	130	127	146	205	150	130	126
117	151	160	181	250	161	134	125
168	170	171	178	183	179	112	124

After zero-bias adjustment

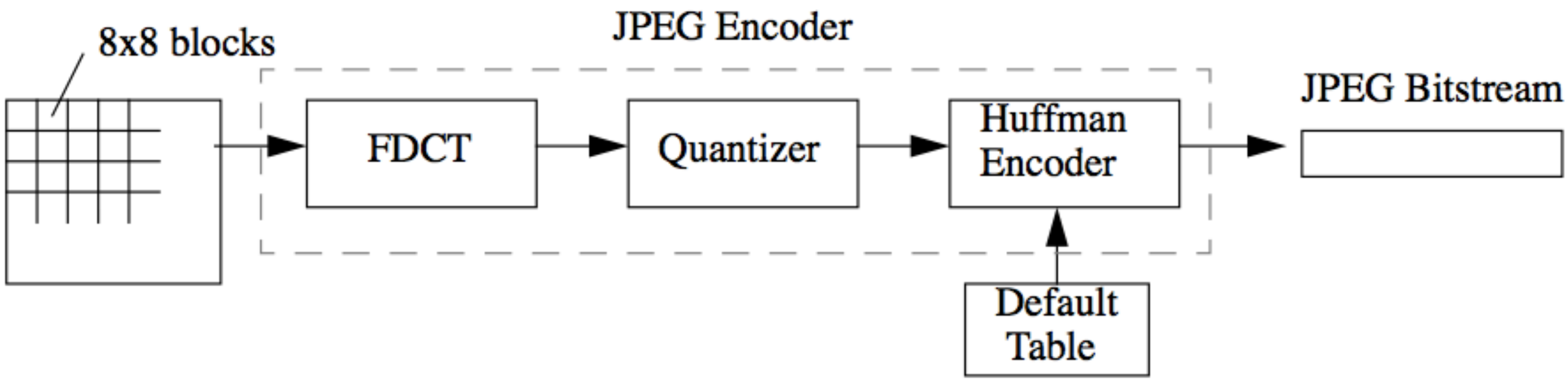
Compression

- Why compress images?
  - Store more images
  - Faster transmission
  - JPEG (Joint Photographic Experts Group)
    - Most popular format; different operational modes
- JPEG core algorithms
  - Image data divided into blocks of 8x8 pixels
  - Three steps for each block
  - 1) DCT (Discrete Cosine Transform) for compression
  - 2) Quantization
  - 3) Huffman encoding

Facts about JPEG Compression

- JPEG stands for Joint Photographic Experts Group
- JPEG compression is used with .jpg and can be embedded in .tiff and .eps files.
  - Used on 24-bit color files.
  - Works well on photographic images.
  - Although it is a lossy compression technique, it yields an excellent quality image with high compression rates.
- The JPEG standard is used in downloading graphics from the internet, in digital cameras, in medical imaging tools, and in many other interesting applications
- We will focus on a small portion of the algorithm-- a section called the Huffman decoder.

JPEG Encoding Flow



Steps in JPEG Compression

- (Optionally) If the color is represented in RGB mode, translate it to YUV.
- Divide the file into 8 X 8 blocks.
- Transform the pixel information from the spatial domain to the frequency domain with the Discrete Cosine Transform.
- Quantize the resulting values by dividing each coefficient by an integer value and rounding off to the nearest integer.
- Look at the resulting coefficients in a zigzag order. Do a run-length encoding of the coefficients ordered in this manner. Follow by Huffman coding.

Example

Example

Example

Apply FDCT

214	224	199	190	222	231	239	236
215	198	179	218	219	223	229	236
202	166	205	215	218	208	144	183
173	177	211	212	200	152	136	196
155	197	205	200	143	133	178	221
158	197	198	153	118	159	202	211
169	192	167	110	139	190	196	199
169	180	131	105	172	193	190	191

$- 128 =$

86	96	71	62	94	103	111	108
87	70	51	90	91	95	101	108
74	38	77	87	90	80	16	55
45	49	83	84	72	24	8	68
27	69	77	72	15	5	50	93
30	69	70	25	−10	31	74	83
41	64	39	−18	11	62	68	71
41	52	3	−23	44	65	62	63

Shift operations

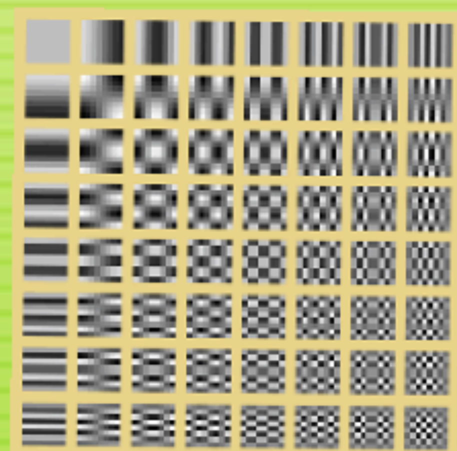
From [0, 255]

To [-128, 127]

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

DCT Result

479	−35	52	−30	−7	−27	10	−3
141	11	−62	11	56	14	5	−5
41	−54	48	98	−35	−19	−14	8
22	−16	56	−48	−48	17	−24	−3
−4	5	−6	−16	14	−15	−15	9
−10	20	−11	9	−7	−15	22	12
−9	12	−12	5	−4	0	14	3
−2	6	−2	3	−3	0	8	−2



Meaning of each position in DCT result-matrix

Quantization

DCT result

479	−35	52	−30	−7	−27	10	−3
141	11	−62	11	56	14	5	−5
41	−54	48	98	−35	−19	−14	8
22	−16	56	−48	−48	17	−24	−3
−4	5	−6	−16	14	−15	−15	9
−10	20	−11	9	−7	−15	22	12
−9	12	−12	5	−4	0	14	3
−2	6	−2	3	−3	0	8	−2

/

Quantization Matrix
(adjustable according to quality)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

=

30	−3	5	−2	0	−1	0	0
12	1	−5	1	2	0	0	0
3	−4	3	4	−1	0	0	0
2	−1	3	−2	−1	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantization result

Archive Step

- Record starting address and image size
  - Can use linked list
- How?
- Example 1
  - Reserve memory for N addresses and N image-size vars
  - Keep a counter for next avail. mem. addr. for next image
  - Initialize addresses and image-size var. to 0
  - As more images come in, update all counters
- Example 2
  - Using linked list
- Tradeoffs?

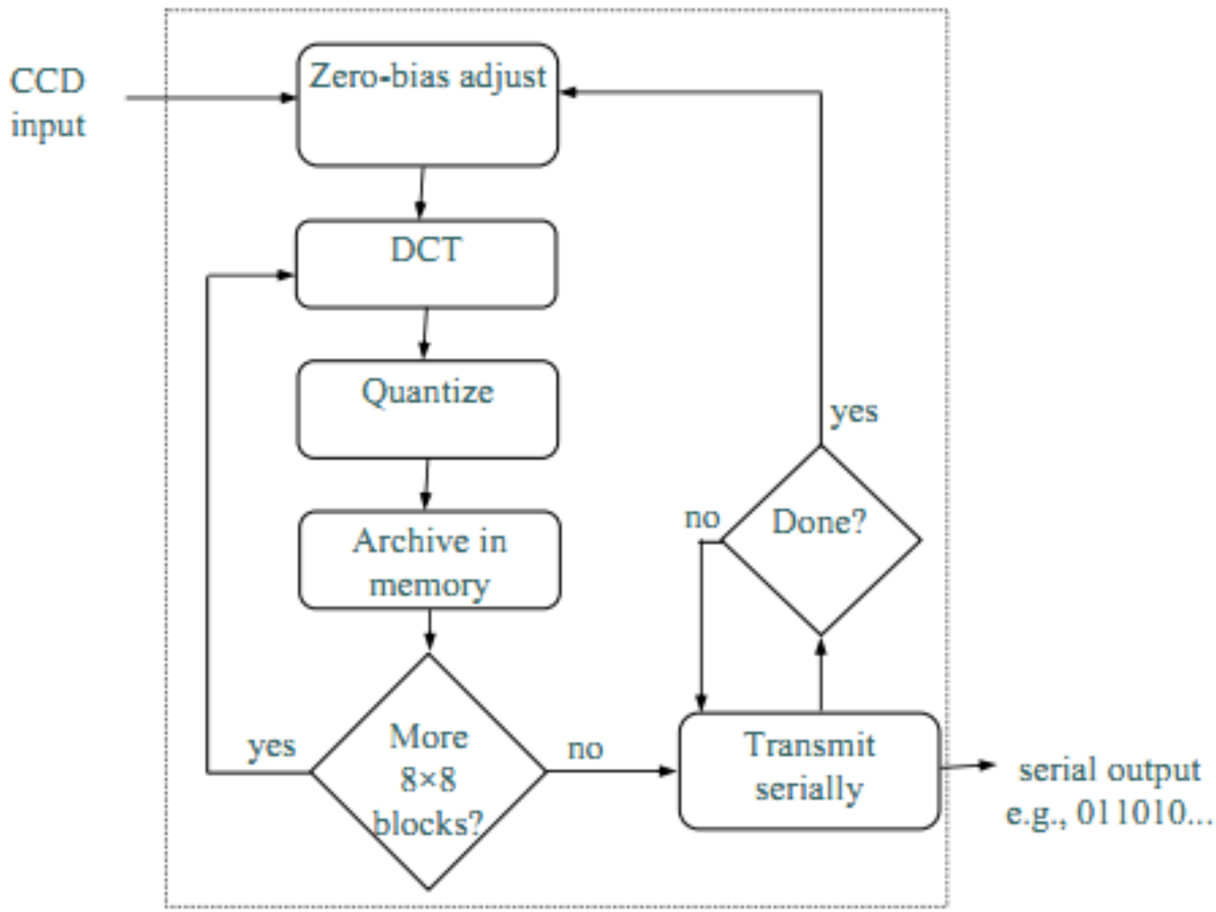
Uploading to PC

- Connected to PC
- UART interface
- Issue commands
  - Read images from camera memory
  - Transmit bits serially using UART
  - While transmitting
    - Reset points
    - Reset image size variables
    - Reset global pointer according

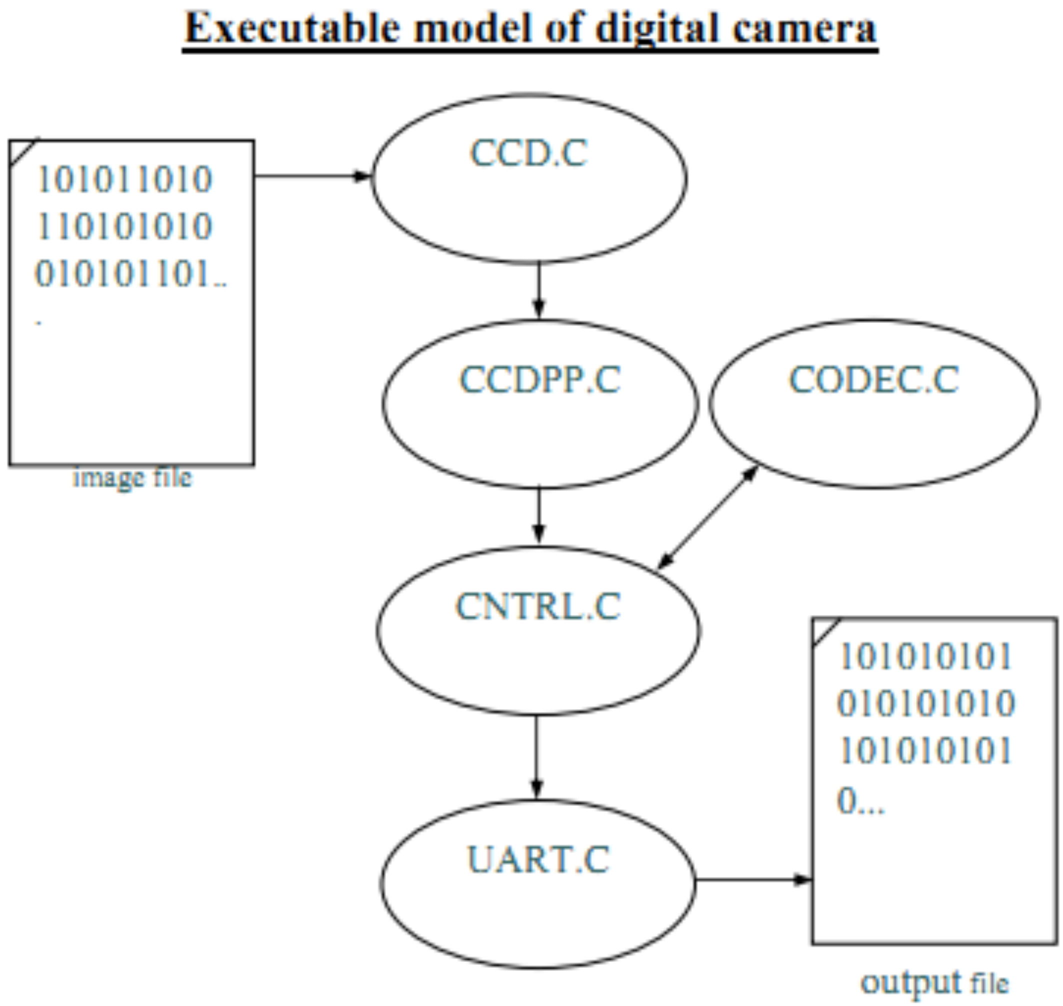
Product Design Requirements

- Typical
  - Performance
  - Size
  - Power
  - Energy
- Design constraint
  - Price threshold, profit margin
- Optimization
  - Do the best
- Optimization vs. Constraints

Functional Spec in Flowchart



Functional Spec in Software Code



Final issues

- Come by my office hours (right after class)
- Any questions or concerns?