
EEL 4783: Hardware/Software Co-design with FPGAs

Lecture 10: Hardware Implementation of CODEC

Prof. Mingjie Lin



Overview

- Huffman Encoding and Decoding
 - Ideas and concepts
 - How

JPEG-Lite

- The input has been modified to 4x4 blocks as opposed to the 8x8 blocks used in the JPEG standard in order to reduce the layout effort of hardware elements
 - A simplified Huffman table will be used by the Encoder and Decoder that contains 10-bit Huffman codes and allows a maximum run-length of 3. The JPEG-baseline standard contains 16-bit Huffman codes and supports a maximum run-length of 15.
 - Only a single AC and a single DC Huffman table will be used. In the JPEG baseline standard, two AC and DC tables were supported.
 - 1-bit Huffman codes are not allowed in JPEG-lite. • Only grayscale images will be decoded in JPEG-lite.
-

Example: a 4x4 block encoded on a bitstream.

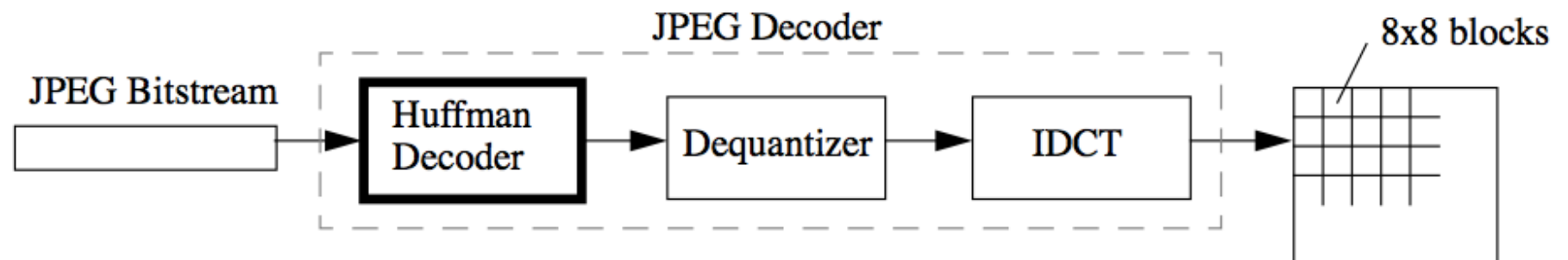
15	0	2	1
1	1	0	0
0	4	0	0
0	0	5	0

DC Coefficient of previous block=12

Run-length	Coeff_size	Huffman Code	Coefficient
0	2	011	11 (3=15-12)
1	2	11011	10 (2)
0	1	00	1 (1)
0	1	00	1 (1)
0	1	00	1 (1)
3	3	111110101	100 (4)
3	0	111001	none
0	3	100	101 (5)
0	0	1010	none - EOB

Bitstream = 0111111011100010010011111101011001110011001011010

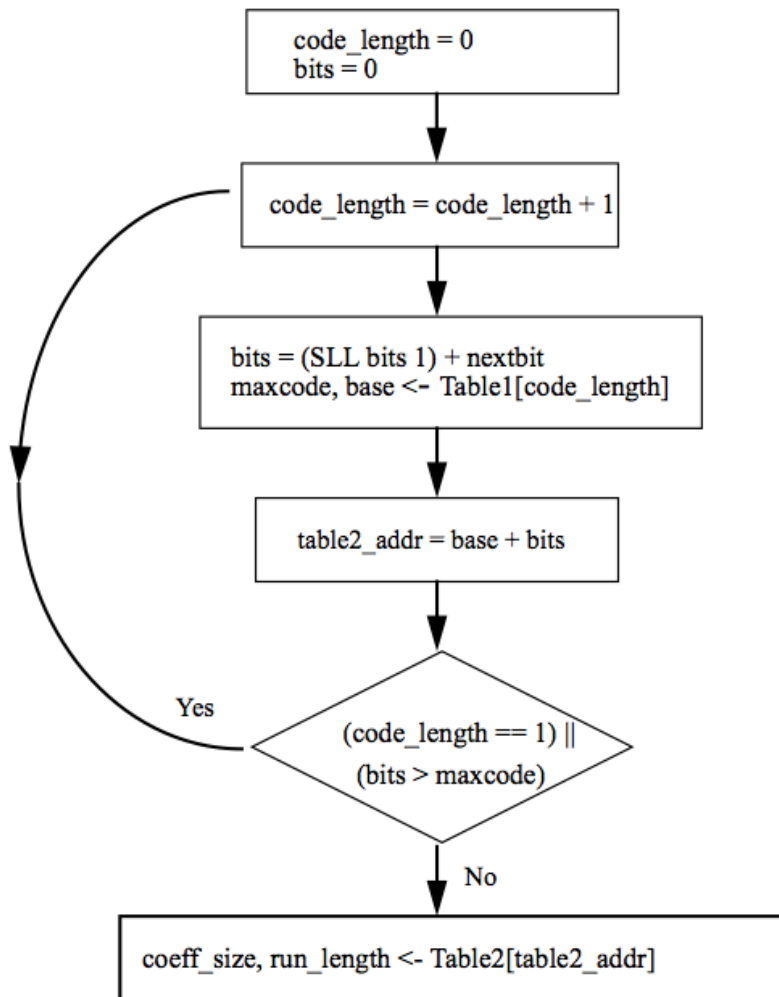
JPEG Decoding Flow



Why Huffman Decoding is hard?

- The difficulty in decoding the bitstream arises in determining how long a Huffman code is (remember that Huffman codes are variable-length)
- Fortunately, the JPEG standard provides a clever algorithm that takes advantage of the way in which Huffman codes are created. All Huffman codes must adhere to the following rule:
 - The most significant $(n-1)$ bits of the smallest Huffman code of length n are greater in value than the largest Huffman code of length $(n-1)$.

Algorithm and Example



Example:

DC Huffman code = 011

code_length	1	2	3
bits	000000000	000000001	000000011
maxcode	undefined	000000000	000000110
base	undefined	000000	111111
table2_addr	undefined	000001	000010
loop condition?	YES	YES	NO
coeff_size	N/A	N/A	2
run_length	N/A	N/A	0

Note that the maxcode and base are undefined when the `code_length == 1`.

Table 1

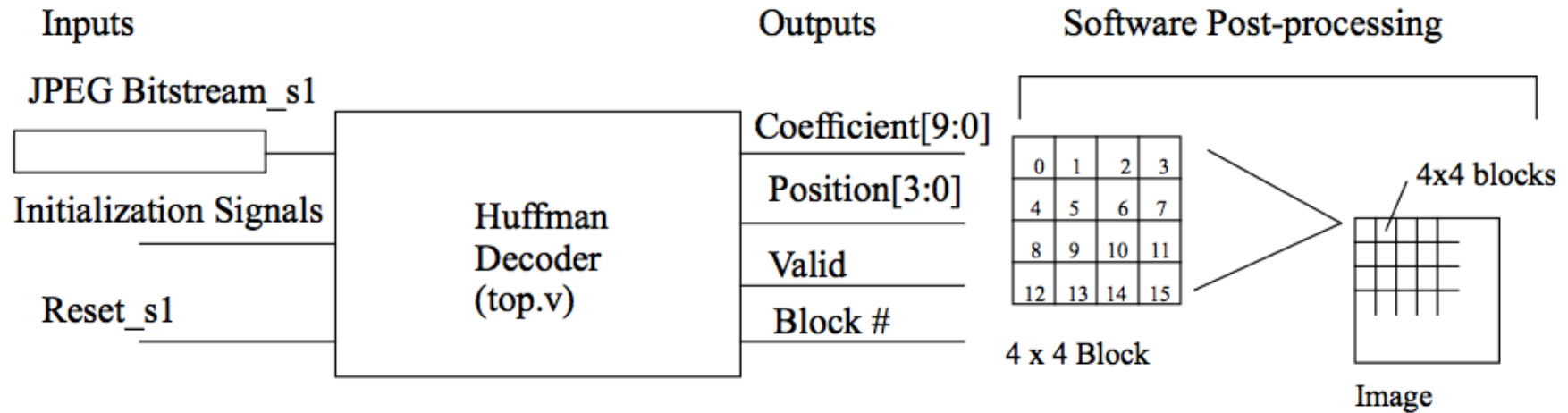
Physical Addr	Code Length	DC Maxcode	DC Base	AC Maxcode	AC Base
0	2	000000000	000000	000000001	001011
1	3	000000110	111111	000000100	001001
2	4	000001110	111000	000001100	000100
3	5	000011110	011000	000011011	110111
4	6	000111110	001010	000111010	011011
5	7	001111110	001011	001110111	100000
6	8	011111110	001100	000000000	000000
7	9	xxxxxxxxx	xxxxxxx	111111100	111000

Table 2

Physical Addr	Huffman Code	Run-length	Coeff Size (Hex)
0	00	0	0
1	010	0	1
2	011	0	2
3	100	0	3
4	101	0	4
5	110	0	5
6	1110	0	6
7	11110	0	7
8	111110	0	8
9	1111110	0	9
10	11111110	0	A

Physical Addr	Huffman Code	Run-length	Coeff Size (Hex)
11	00	0	1
12	01	0	2
13	100	0	3
14	1010 (EOB)	0	0
15	1011	0	4
16	1100	1	1
17	11010	0	5
18	11011	1	2
19	111000	2	1
20	111001	3	0
21	111010	3	1
22	1110110	0	6
23	1110111	1	3
24	111100000	0	7
25	111100001	0	8
26	111100010	0	9
27	111100011	0	A
28	111100100	1	4
29	111100101	1	5
30	111100110	1	6

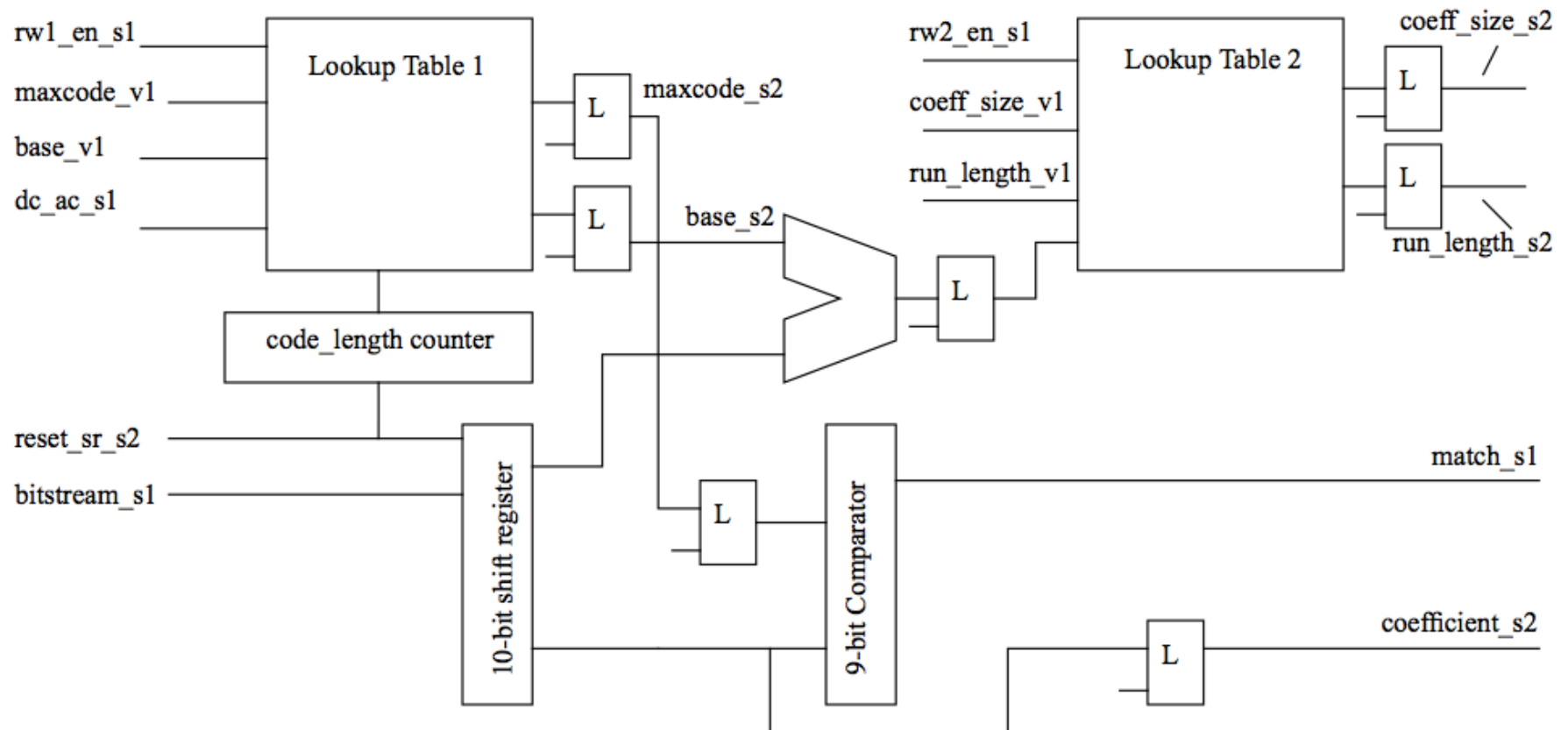
The Big Picture



Control.v

- Explain

Datapath Block Diagram



datapath.v

- Explain

Conclusion

- FPGAs with embedded processors can simplify design of instrument interfaces
- Starting with an evaluation board decreases development time
- Flexible physical implementation
 - Use evaluation board as is
 - Modify design files
 - Build custom design around selected FPGA

Final issues

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