



# Digital Camera Design in SpecC

---

Aseem Gupta

*Dept. of Electrical Engineering & Computer  
Science*

*University of California, Irvine*



**UCIrvine**  
University of California, Irvine



# Outline

---

No Time for this Today!

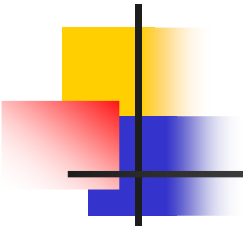




# Goals

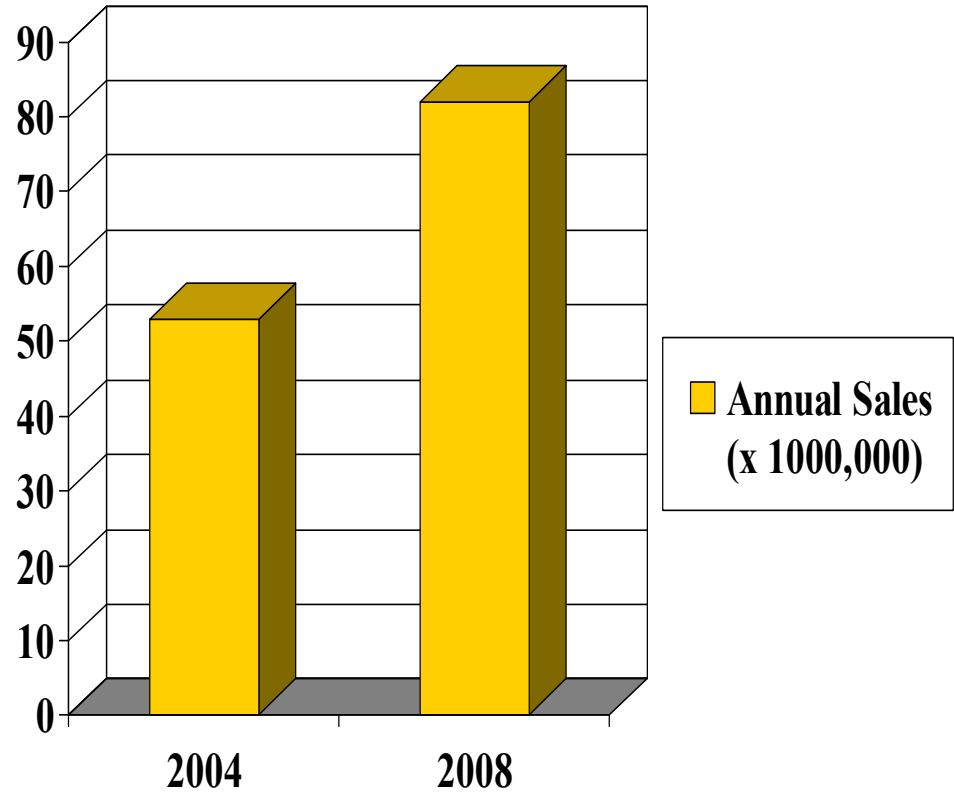
---

- Use SpecC for design of Digital Camera
- Refine behavior
- Validate
- Architectural Exploration
- Profile

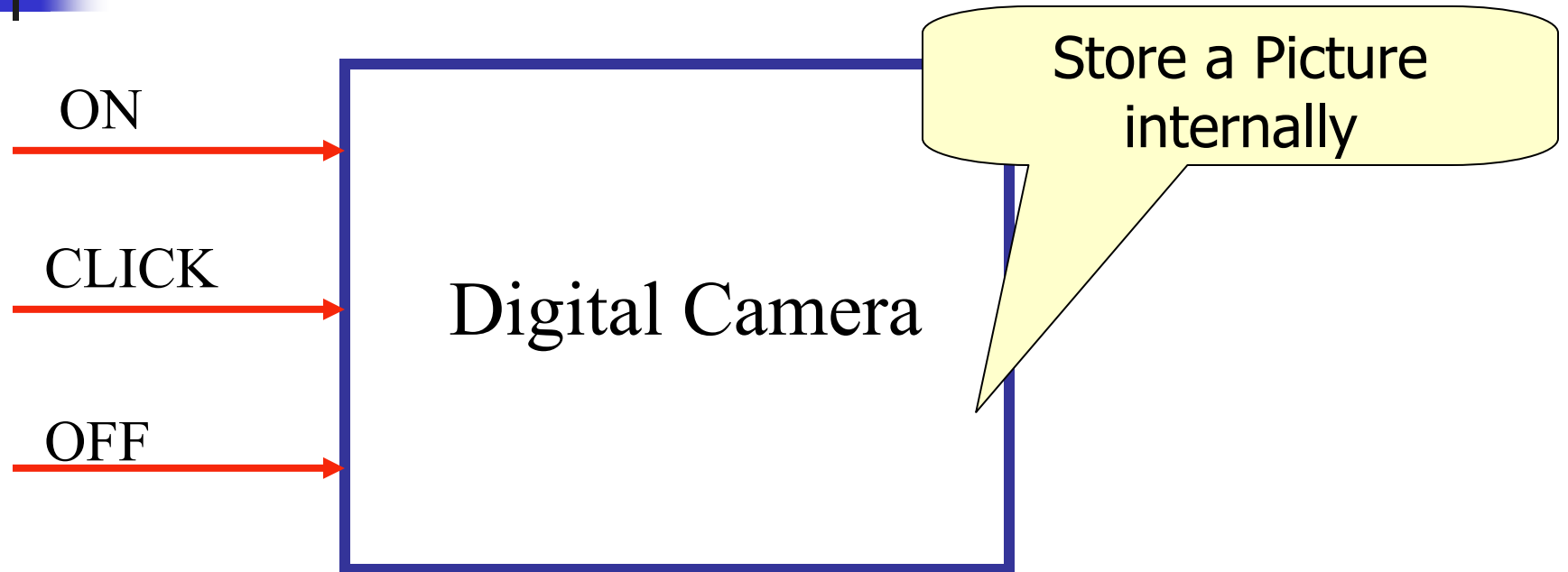


# Digital Camera

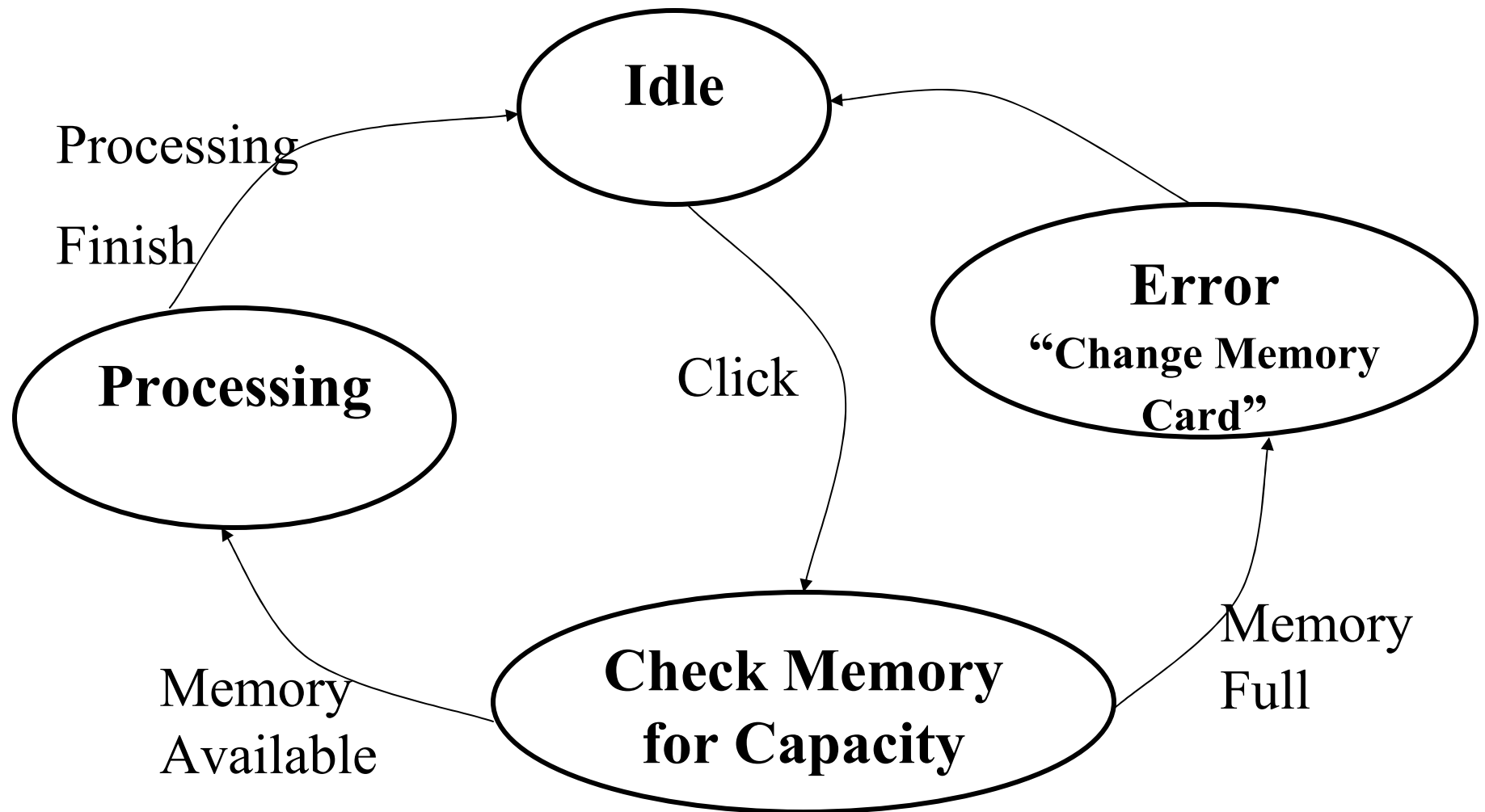
## Sales



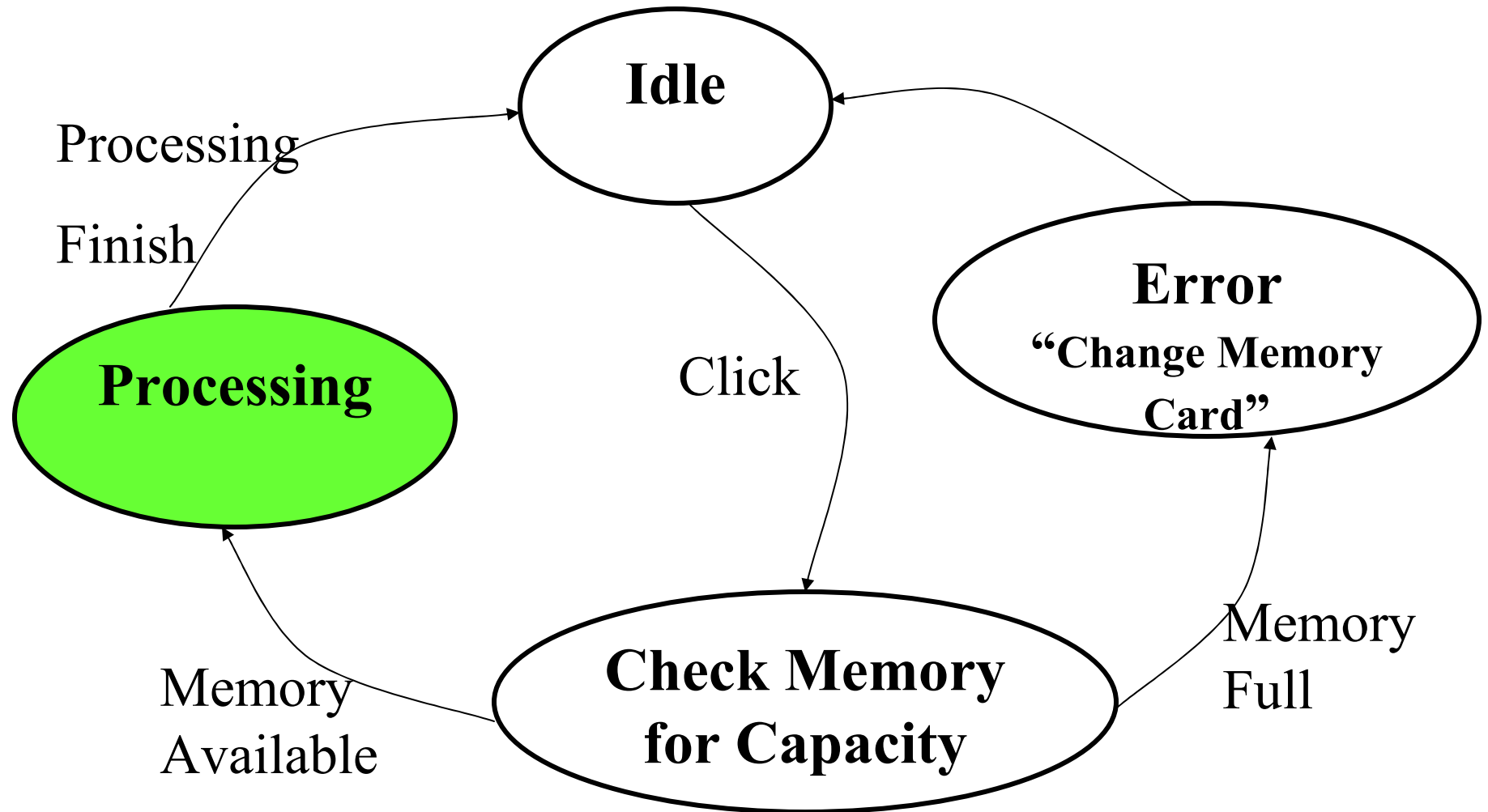
# User's Perspective - Digital Camera



# FSMD (Top Level) - Digital Camera

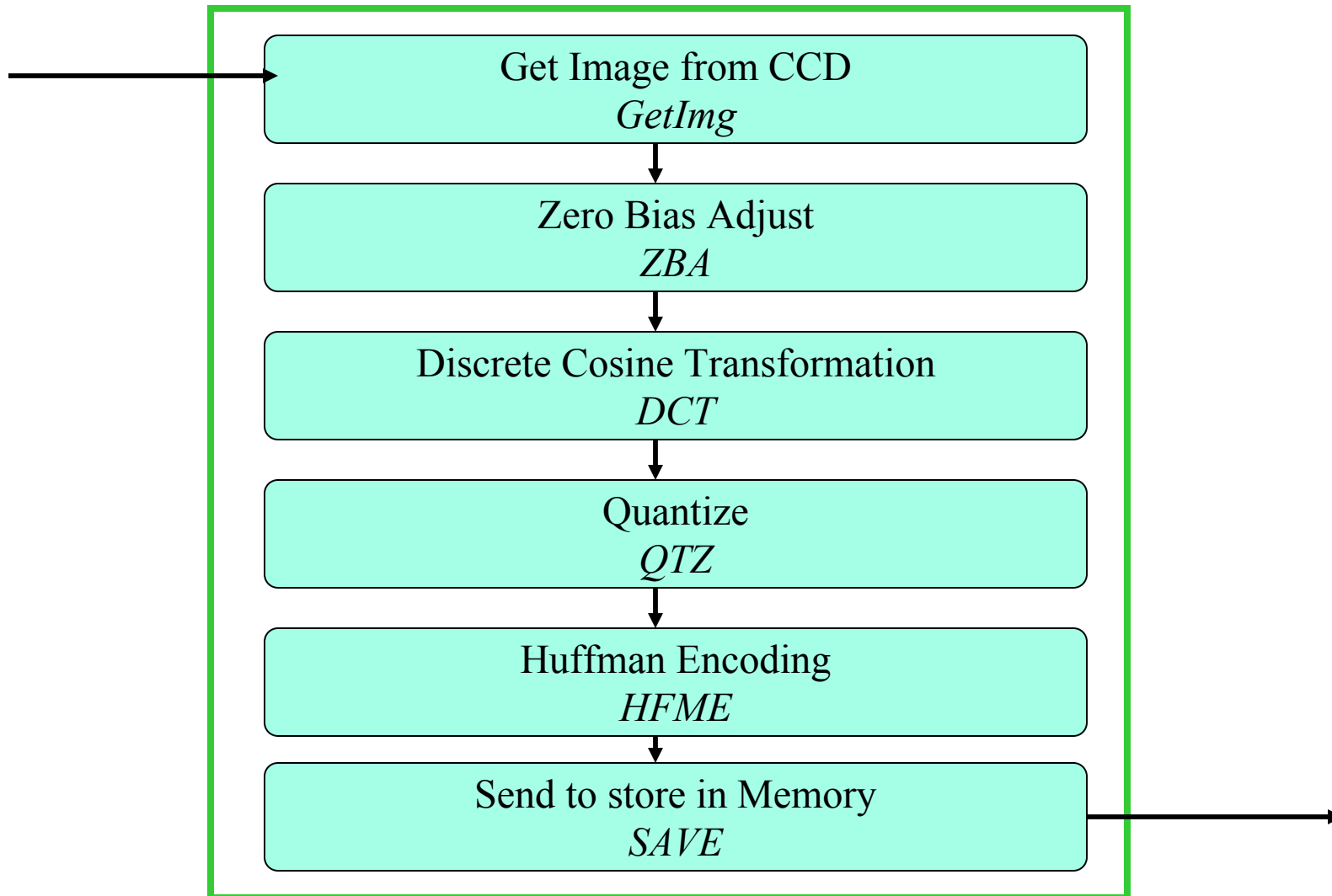


# FSMD (Top Level) - Digital Camera





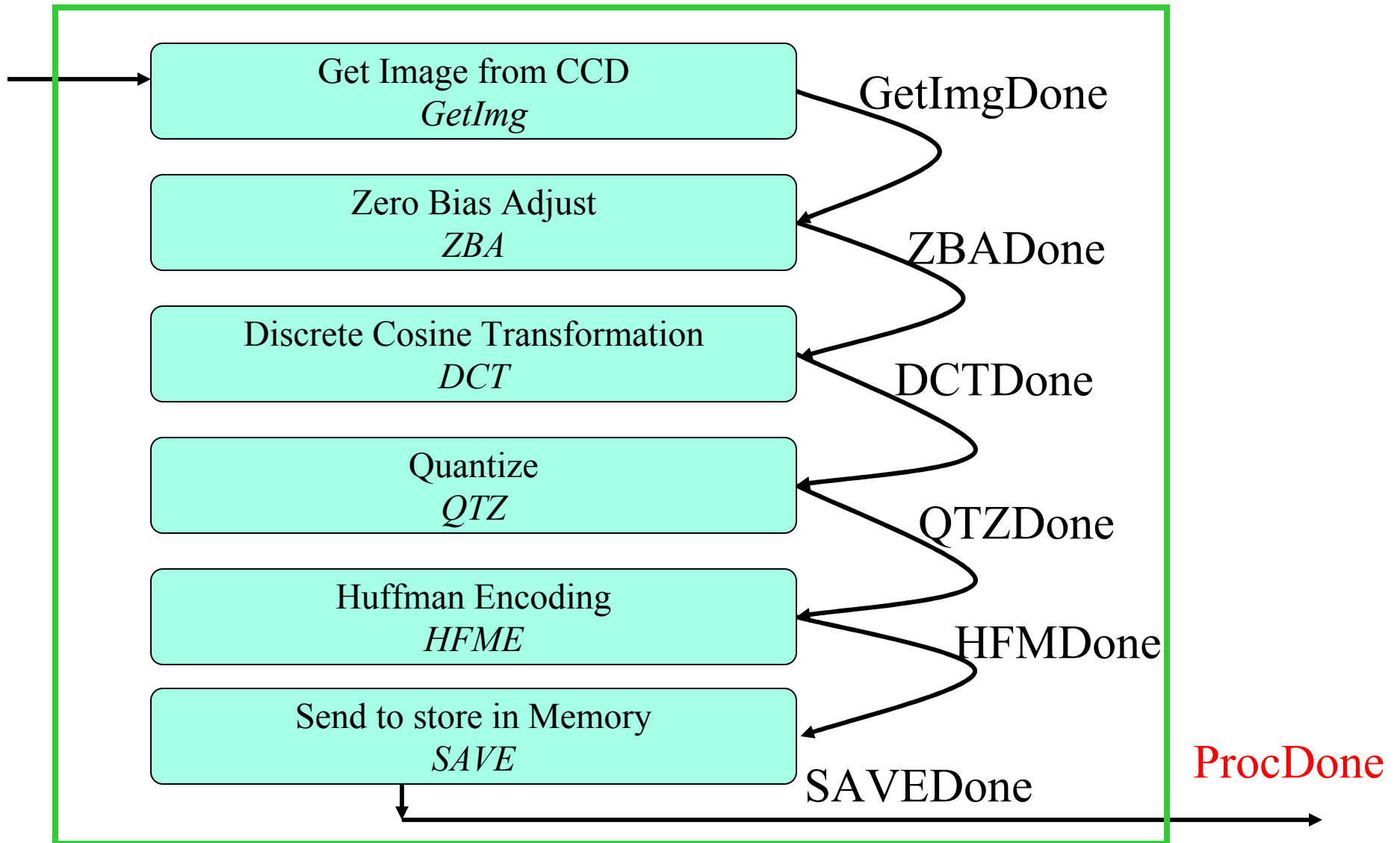
# Processing





# Processing

as FSM



# Charge-coupled device (CCD)

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

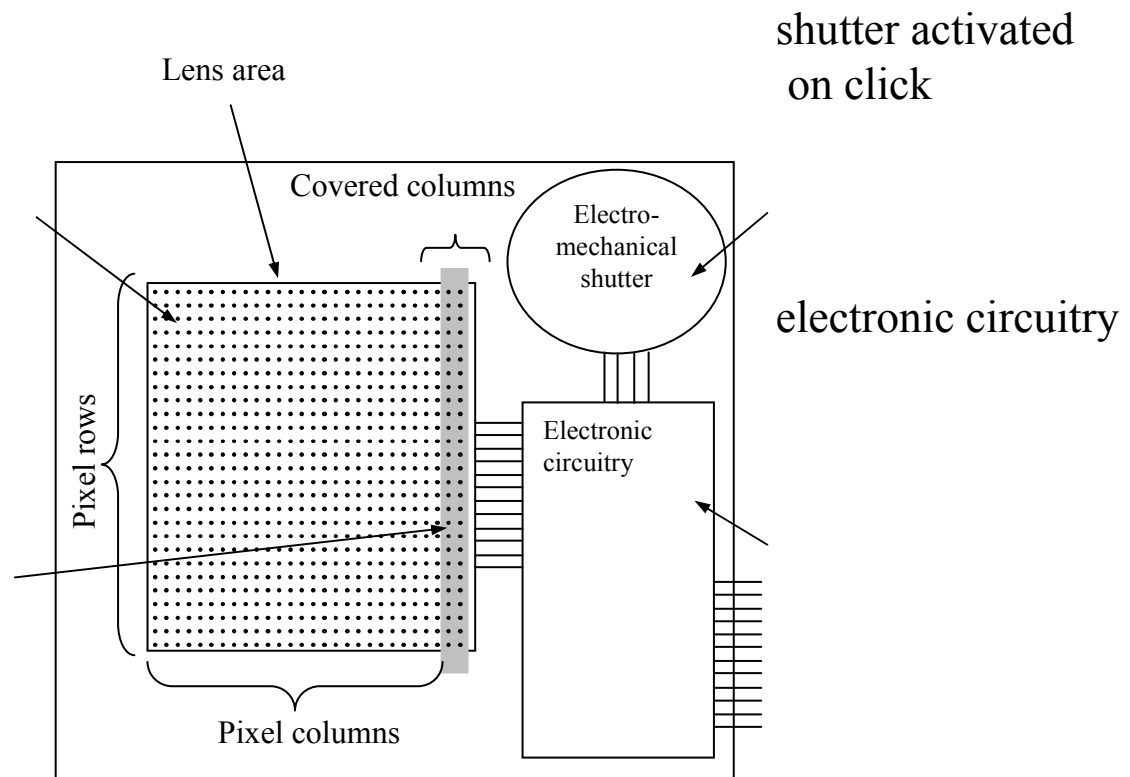
electrical charged

converted to a 8-bit value.

0: no exposure

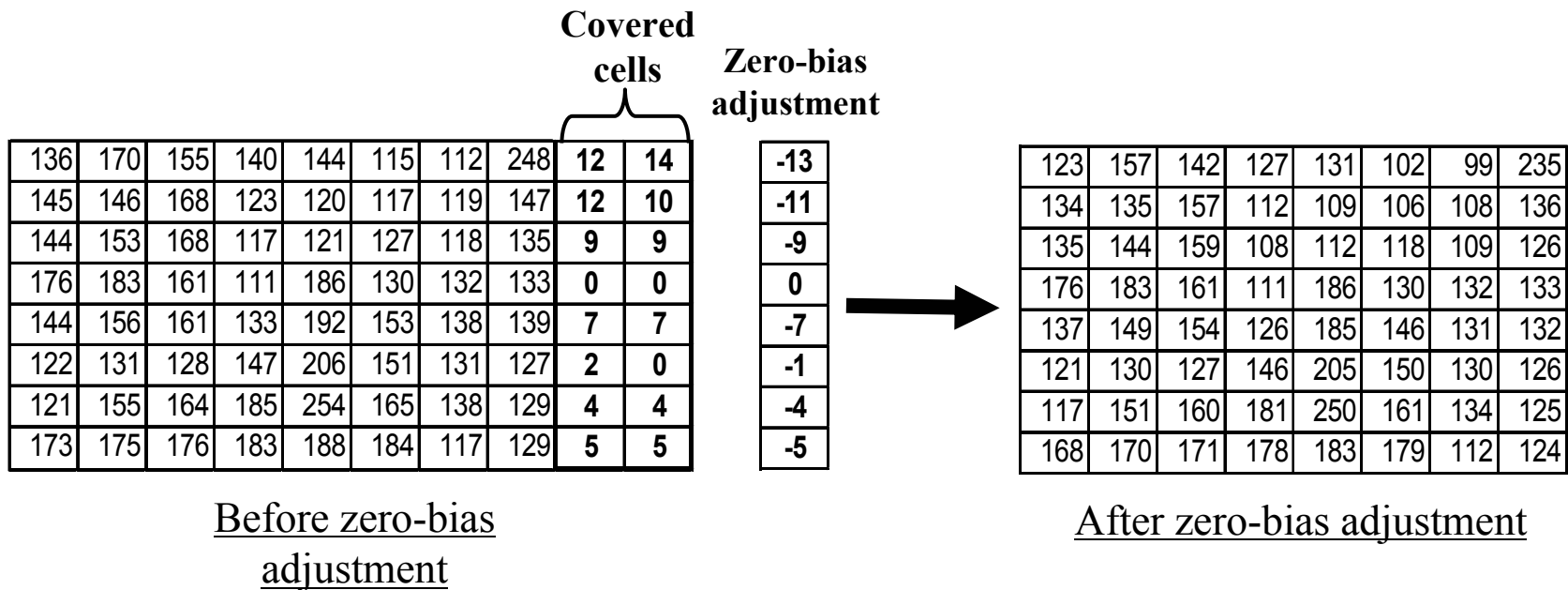
255: intense exposure

Columns covered with black strip of paint for zero-bias adjustments of whole row.



# Zero-bias error

- Cells: Inaccurate response to light
- Left most columns blacked out:
  - Reading of other than 0 in blocked cells is zero-bias error
  - Each row is corrected by subtracting the average error found in blocked cells for that row





## DCT step

---

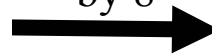
- Transforms original 8 x 8 block into a cosine-frequency domain
- FDCT (Forward DCT) formula
  - $C(h) = \text{if } (h == 0) \text{ then } 1/\sqrt{2} \text{ else } 1.0$ 
    - Auxiliary function used in main function  $F(u,v)$
  - $F(u,v) = 1/4 \times C(u) \times C(v) \sum_{x=0..7} \sum_{y=0..7} D_{xy} \times \cos(\pi(2u + 1)u/16) \times \cos(\pi(2y + 1)v/16)$ 
    - Gives encoded pixel at row  $u$ , column  $v$
    - $D_{xy}$  is original pixel value at row  $x$ , column  $y$

# Quantization step

- compression @ expense of image quality
  - Reduce bit precision, fewer bits needed for encoding
    - One way is to divide all values by a factor of 2
      - Simple right shifts can do this

1150	39	-43	-10	26	-83	11	41
-81	-3	115	-73	-6	-2	22	-5
14	-11	1	-42	26	-3	17	-38
2	-61	-13	-12	36	-23	-18	5
44	13	37	-4	10	-21	7	-8
36	-11	-9	-4	20	-28	-21	14
-19	-7	21	-6	3	3	12	-21
-5	-13	-11	-17	-4	-1	7	-4

Divide each  
cell's value  
by 8



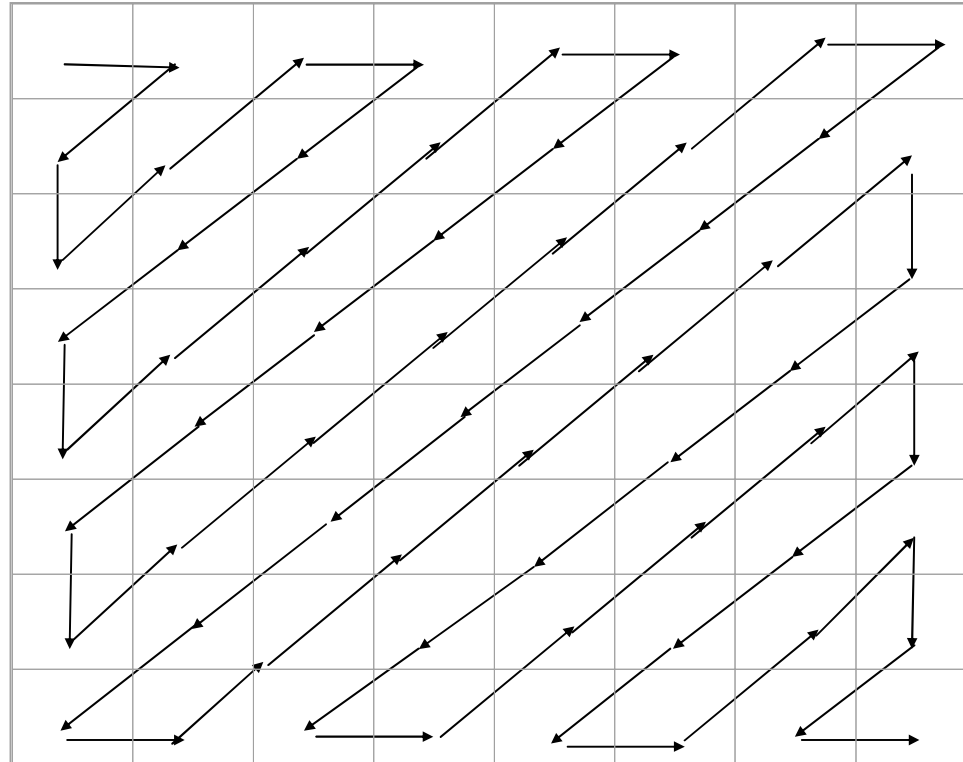
144	5	-5	-1	3	-10	1	5
-10	0	14	-9	-1	0	3	-1
2	-1	0	-5	3	0	2	-5
0	-8	-2	-2	5	-3	-2	1
6	2	5	-1	1	-3	1	-1
5	-1	-1	-1	3	-4	-3	2
-2	-1	3	-1	0	0	2	-3
-1	-2	-1	-2	-1	0	1	-1

After being decoded using  
DCT

After quantization

# Huffman encoding step

- Serialize 8 x 8 block of pixels
  - Values are converted into single list using zigzag pattern

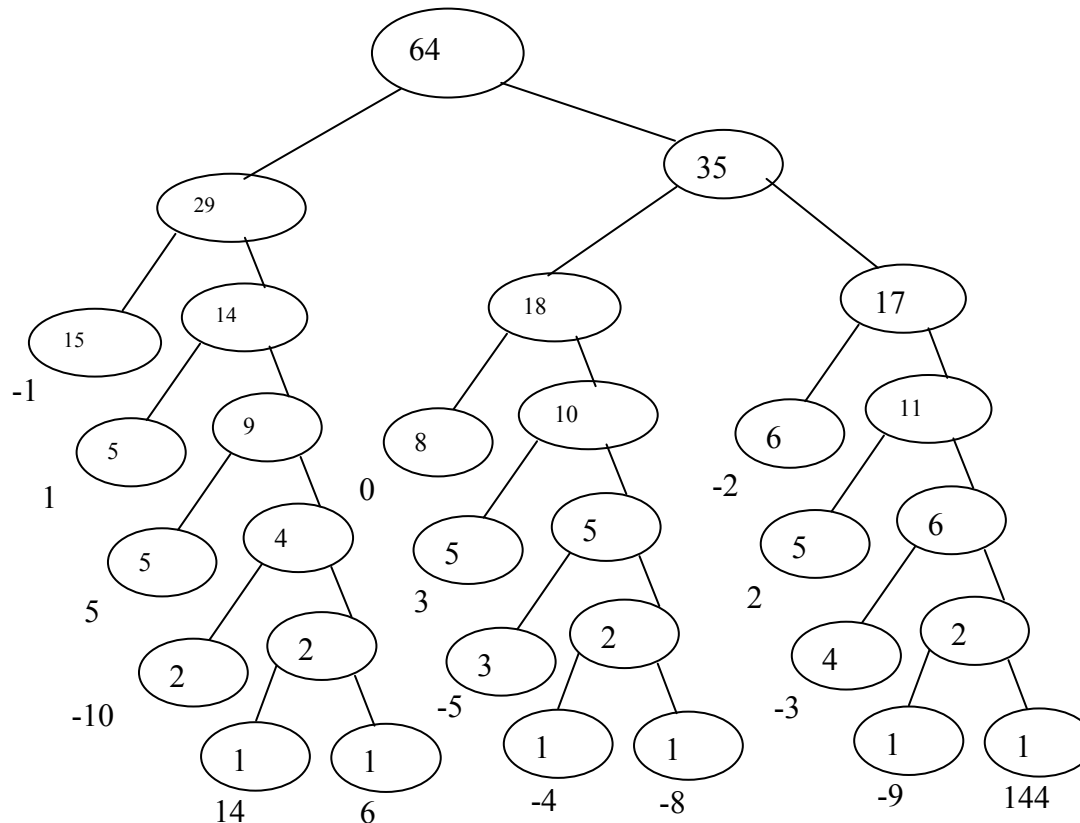


# Huffman encoding example

Pixel frequencies

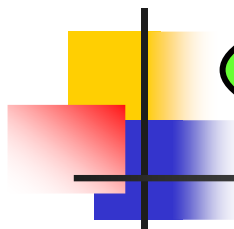
-1	15x
0	8x
-2	6x
1	5x
2	5x
3	5x
5	5x
-3	4x
-5	3x
-10	2x
144	1x
-9	1x
-8	1x
-4	1x
6	1x
14	1x

Huffman tree



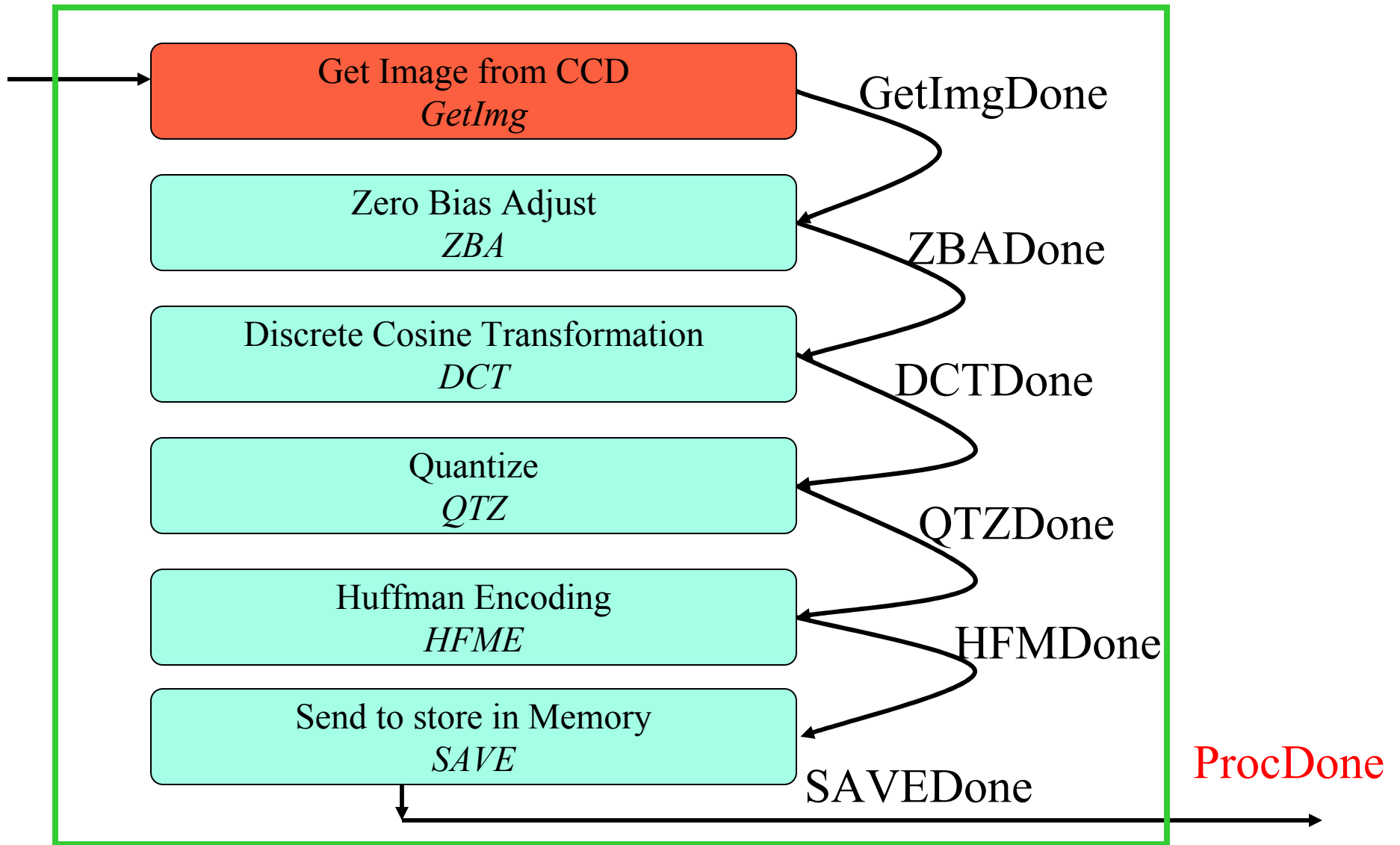
Huffman codes

-1	00
0	100
-2	110
1	010
2	1110
3	1010
5	0110
-3	11110
-5	10110
-10	01110
144	111111
-9	111110
-8	101111
-4	101110
6	011111
14	011110

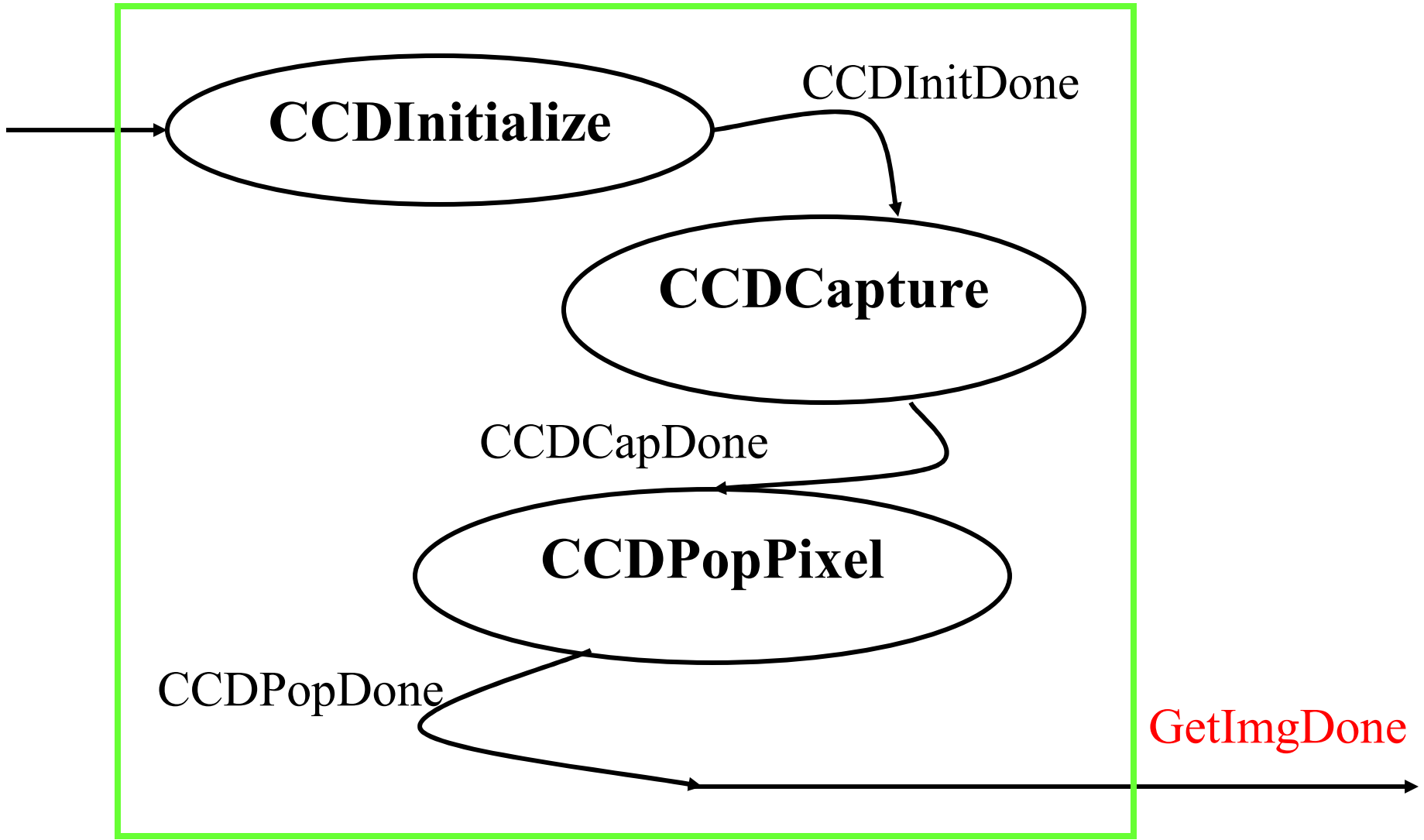


# Processing

as FSM

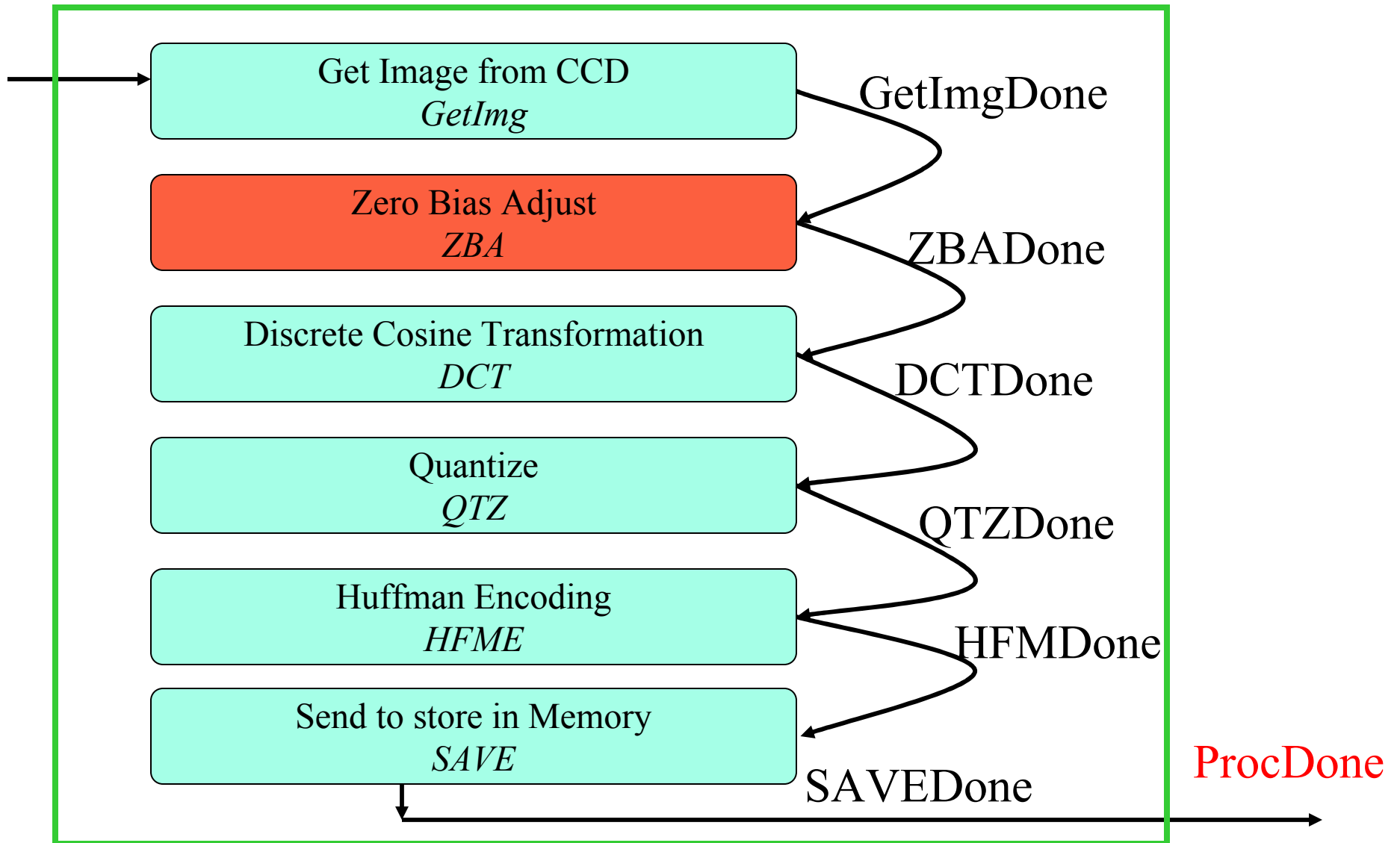


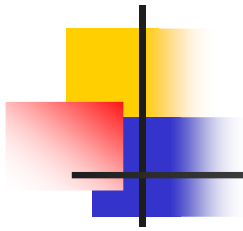




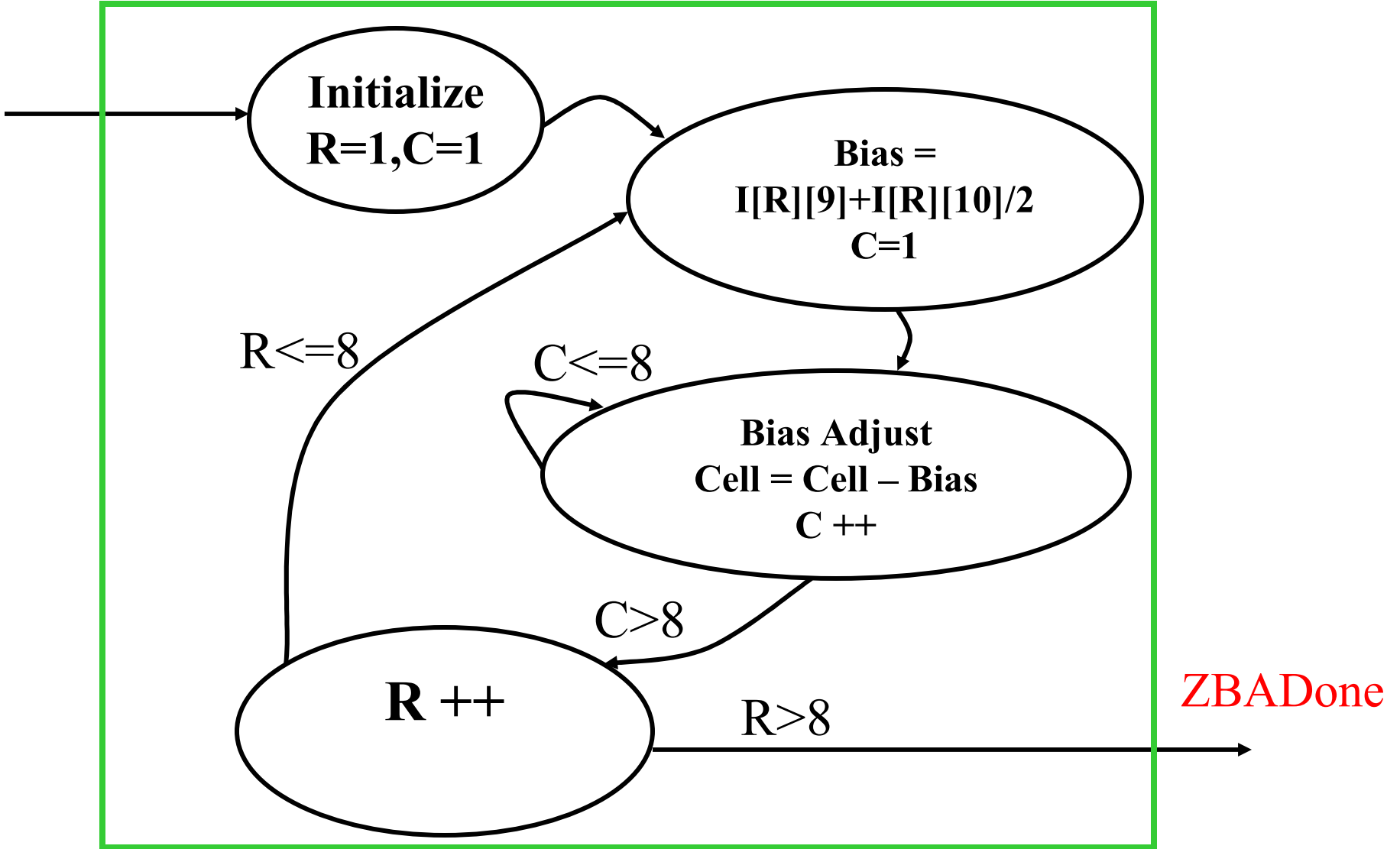
# Processing

as FSM



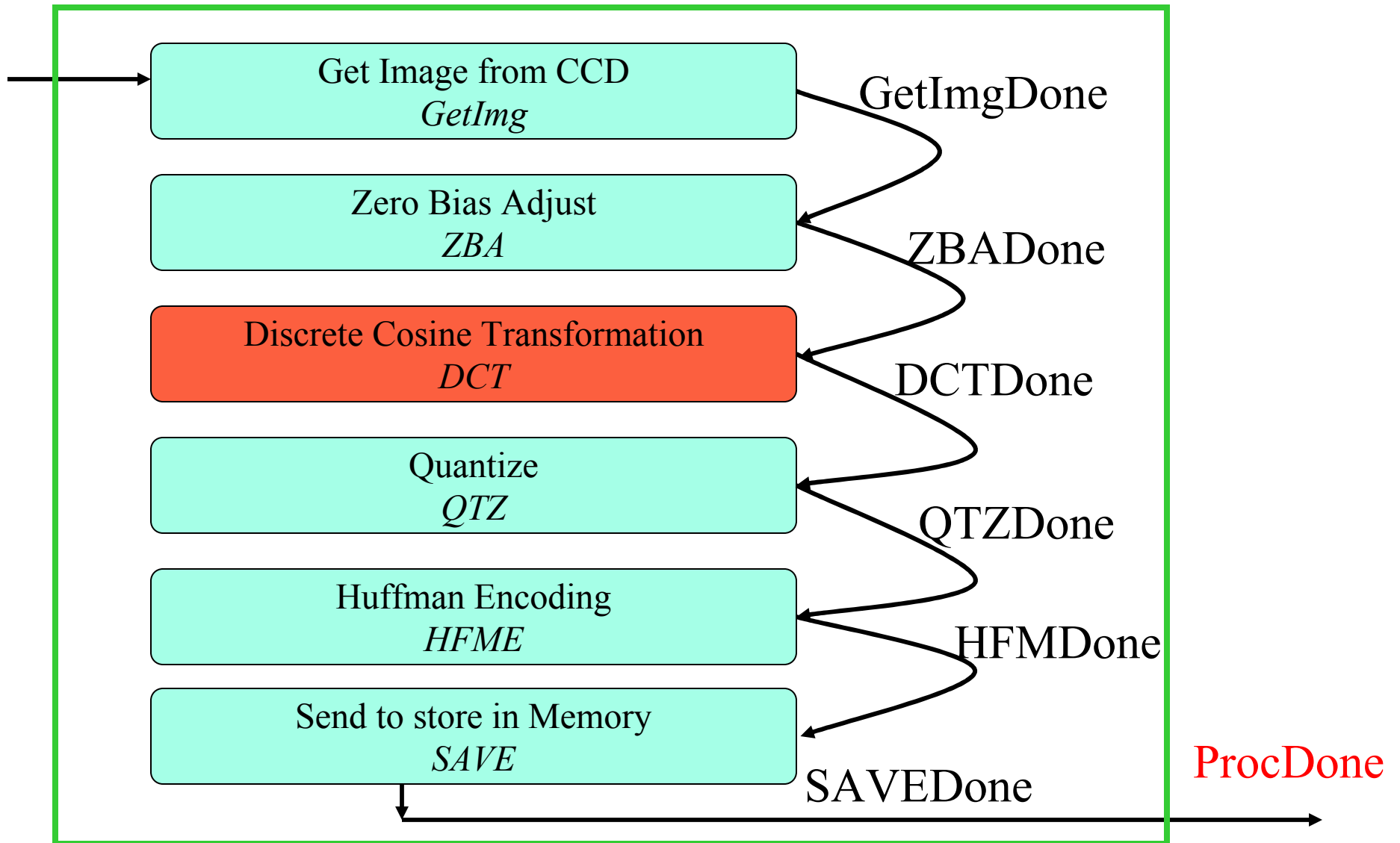


Zero Bias Adjust  
*ZBA*



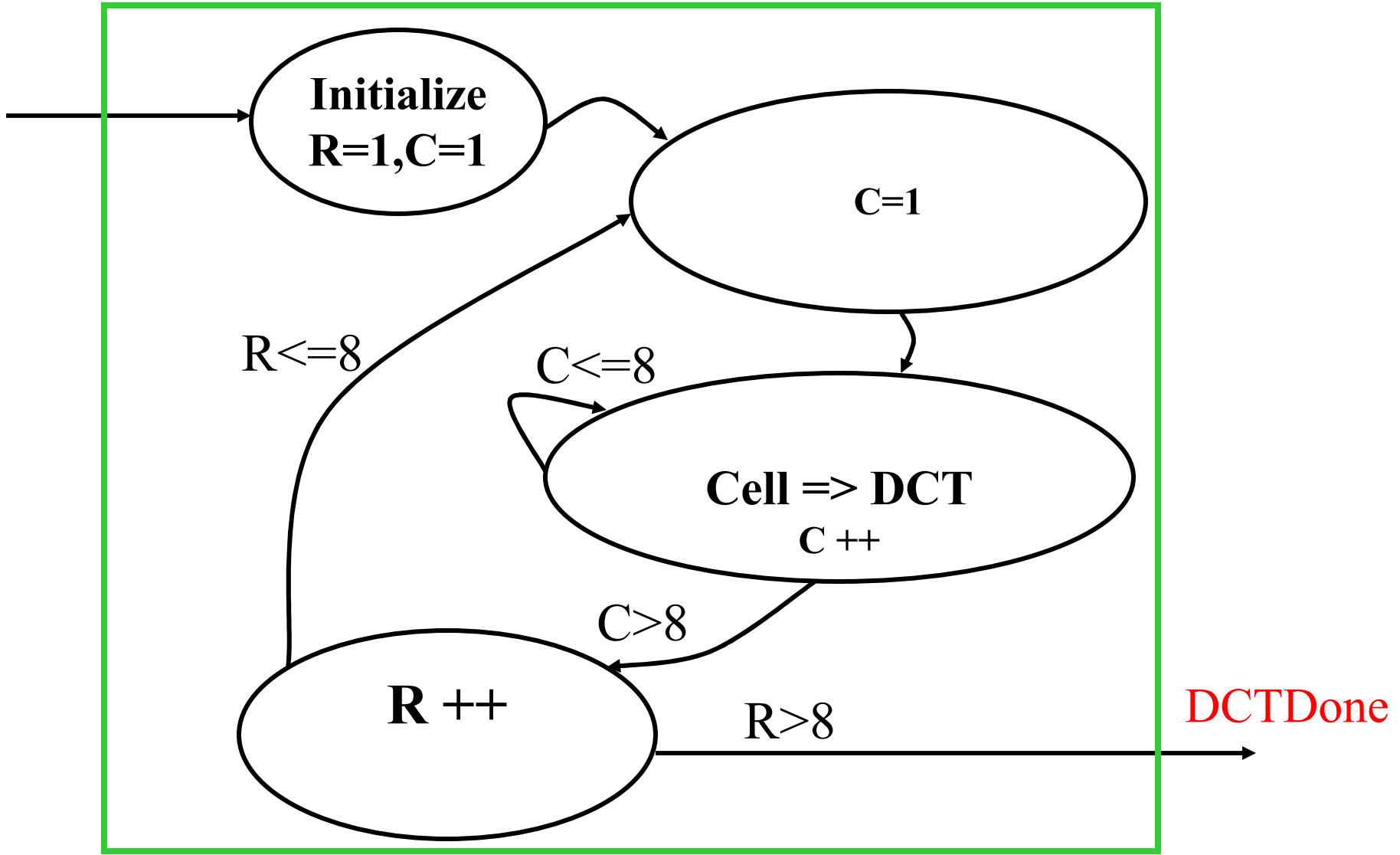
# Processing

as FSM



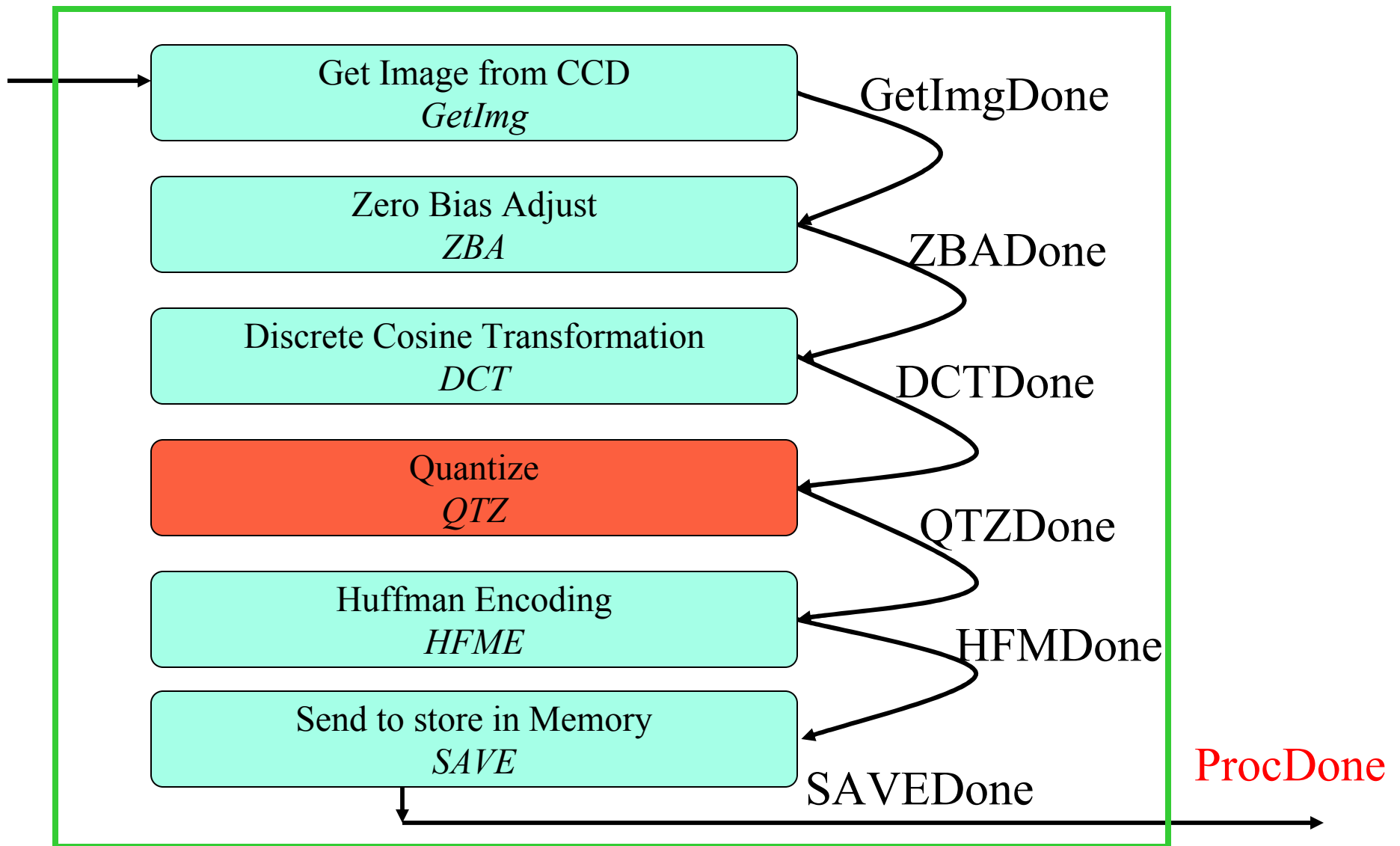


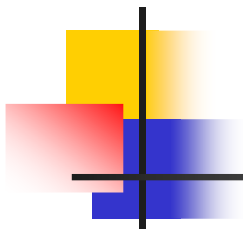
Discrete Cosine Transform.  
*DCT*



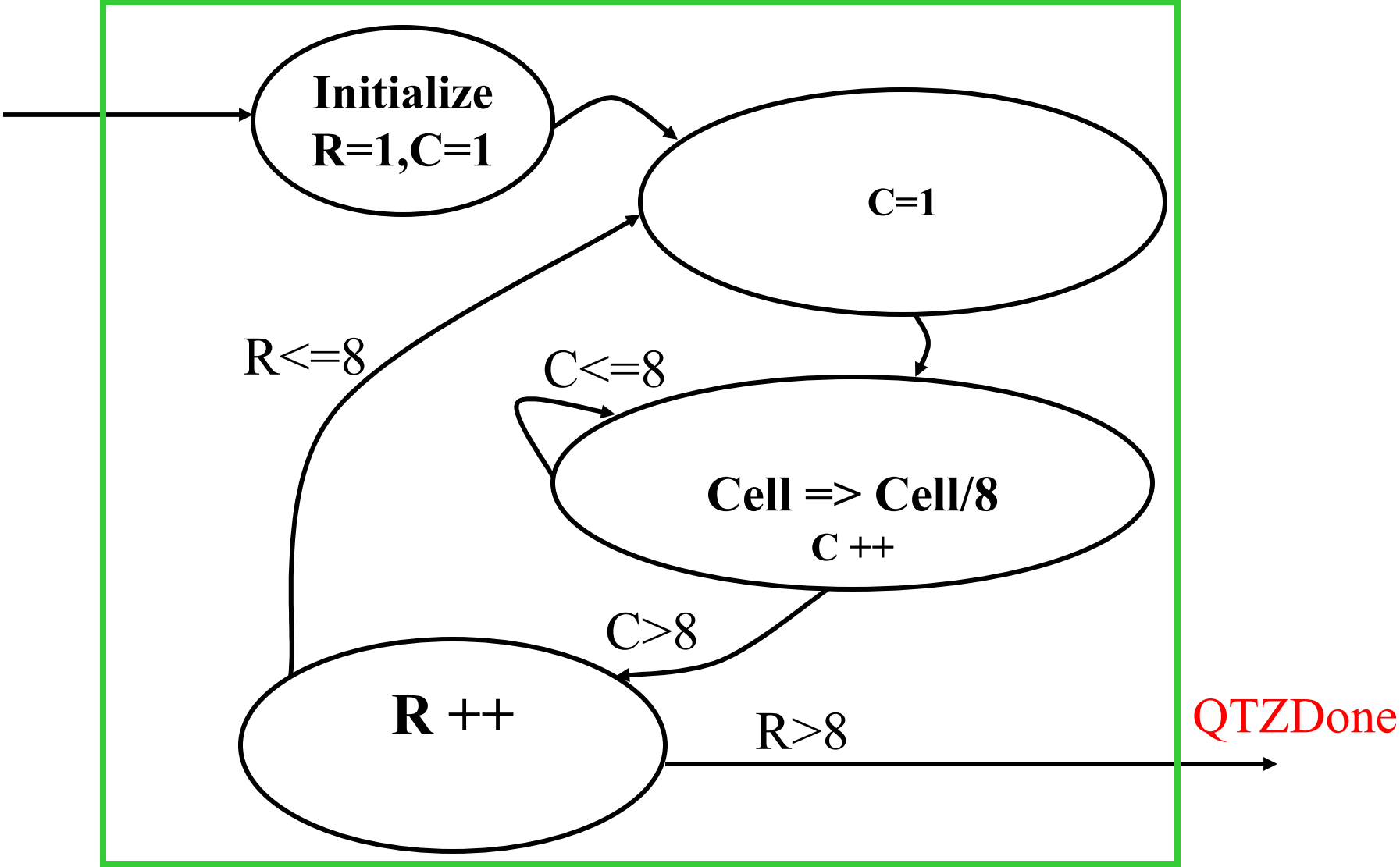
# Processing

as FSM



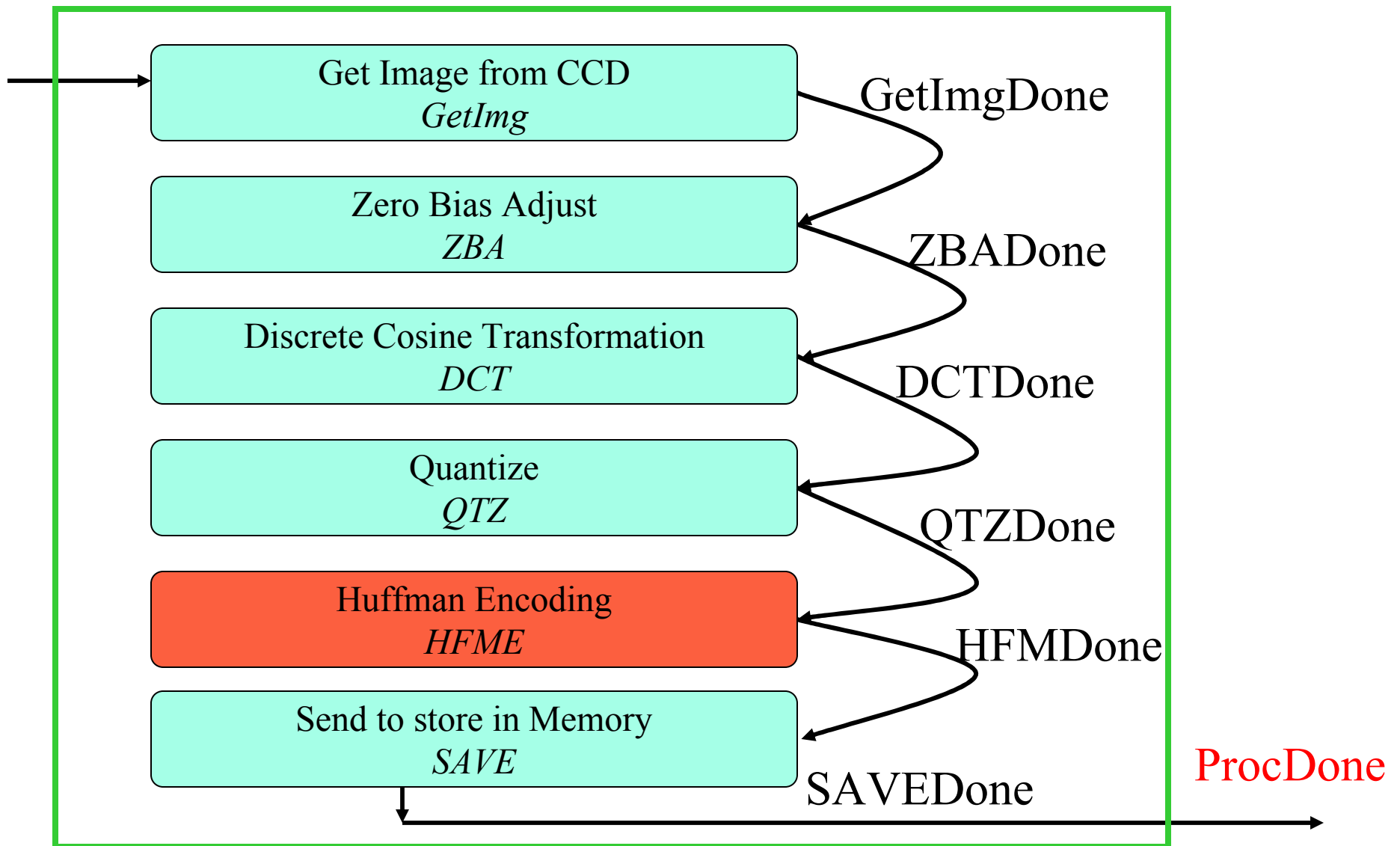


Quantization  
*QTZ*

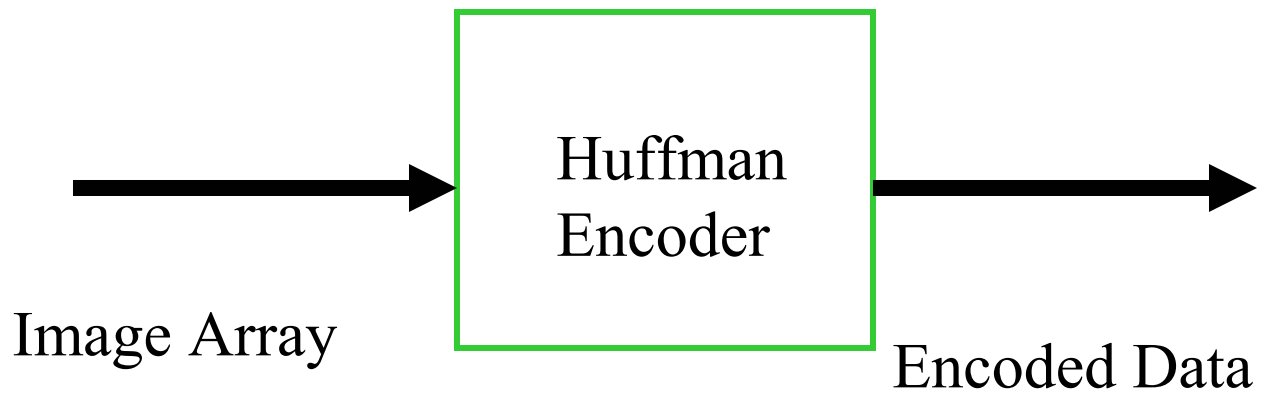


# Processing

as FSM

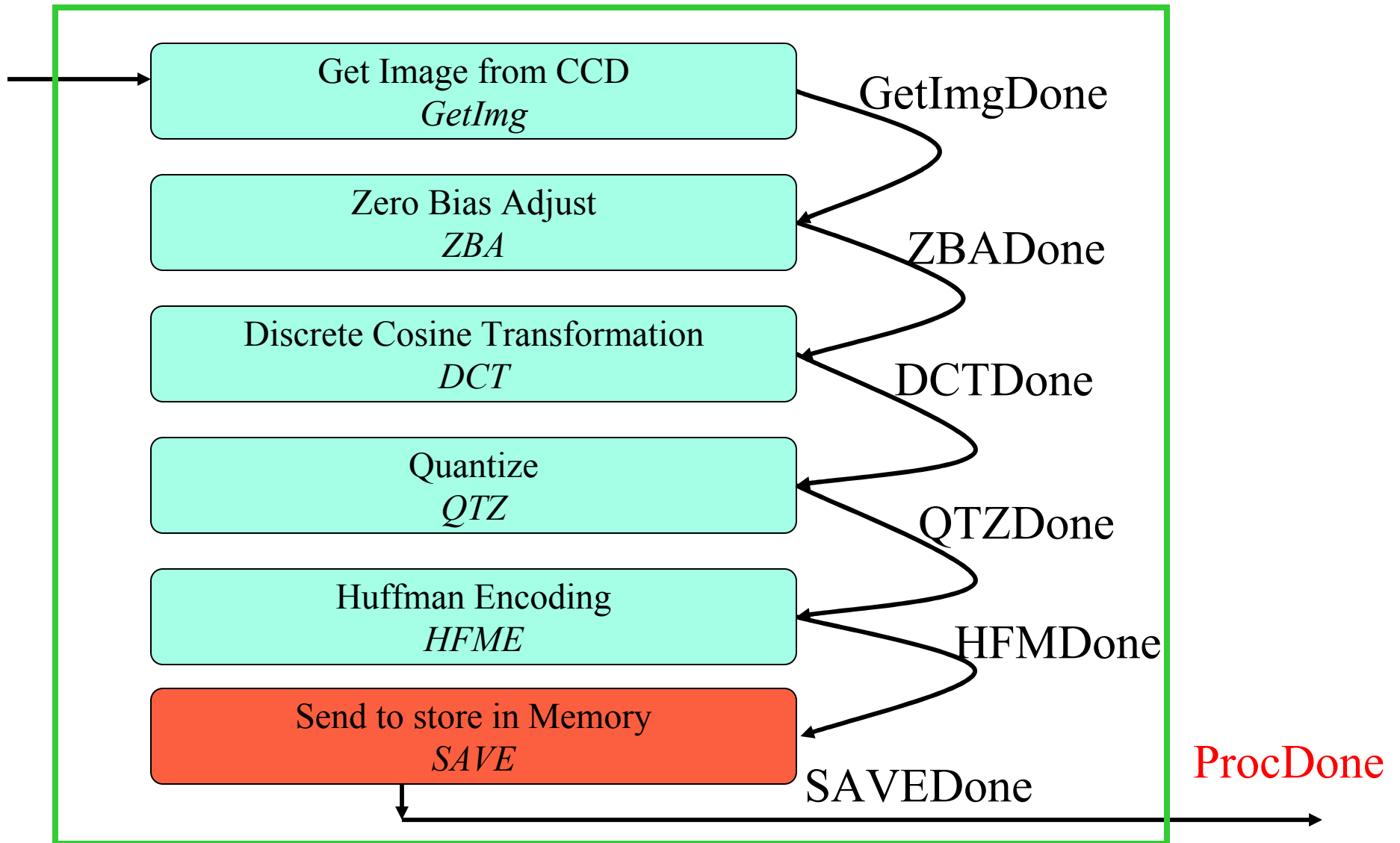


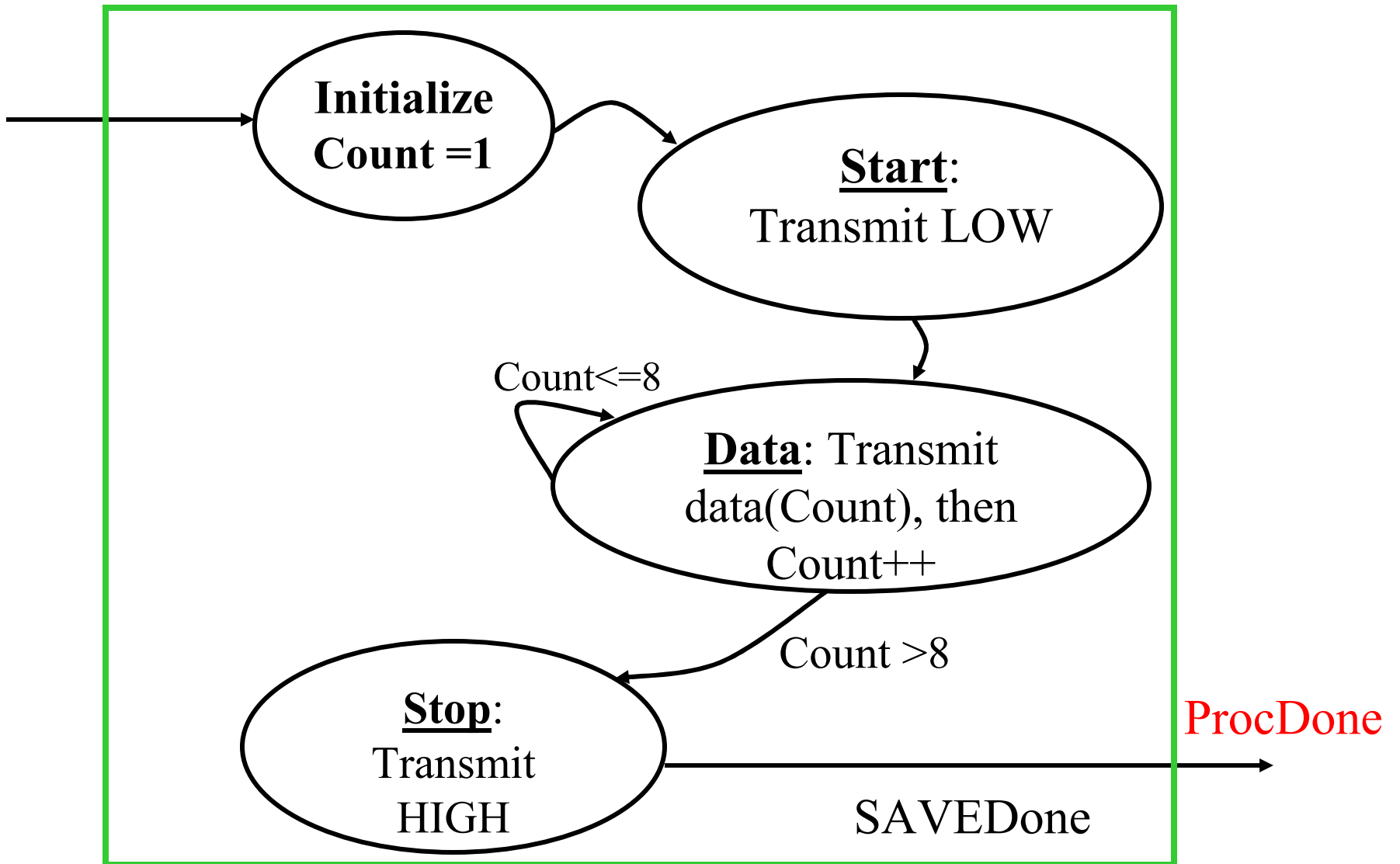




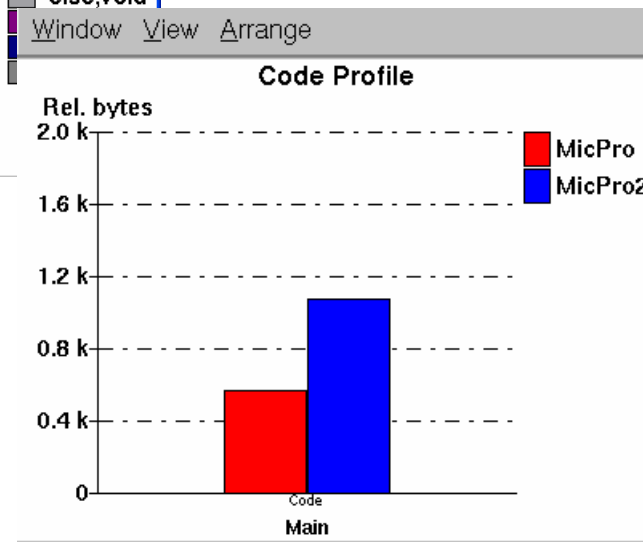
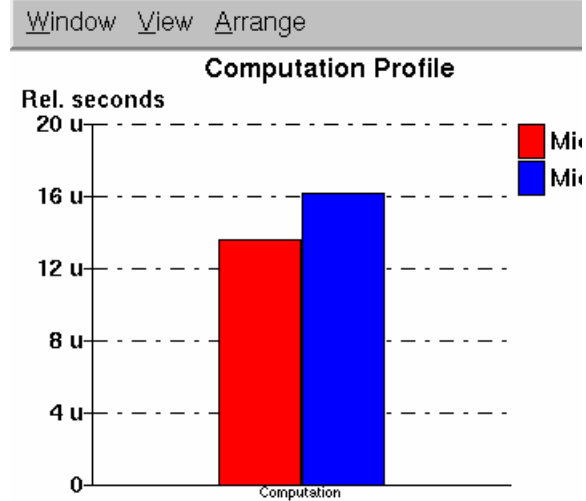
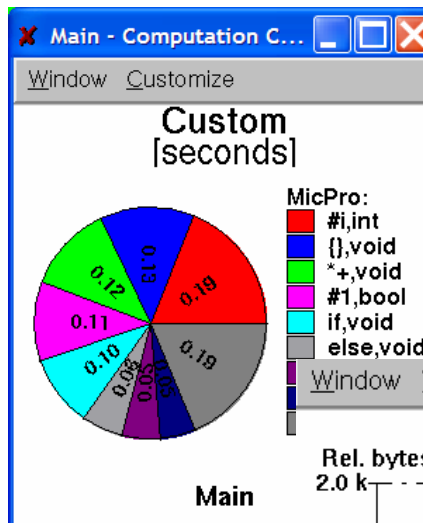
# Processing

as FSM





# Demo - SCE





# Multi-Criteria Decision Making

---

The models in this stream

- Focus on problems with discrete decision spaces.
- Use approaches from discrete mathematics
- Do not try to compute an optimal solution, but try to determine a ranking of the alternatives with respect to several criteria, or they try to find the “optimal” actions amongst the existing alternatives



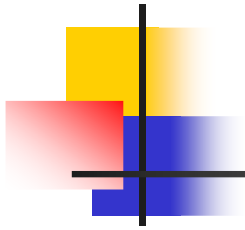
# Derivation of Rankings using different MCDM methods

---

## Using WSM Method

$$M^*_{\text{WSM-score}} = \min_i \sum_{j=1}^n a_{ij} w_j \text{ for } i = 1, 2, 3, \dots, m.$$

	<b>WSM Score</b>	<b>Rankings</b>
<b>M1</b>		1
<b>M2</b>		2
<b>M3</b>		3
<b>M4</b>		4
<b>M5</b>		5
<b>M6</b>		6
<b>M7</b>		7



# Derivation of Rankings using different MCDM methods

## Using WPM Method

$$R(M_K/M_L) = \prod_{j=1}^n (a_{Kj} / a_{Lj})^{w_j}$$

	WPM Score		WPM Score		WPM Score
<b>R(M1/M2)</b>		<b>R(M2/M3)</b>		<b>R(M4/M5)</b>	
<b>R(M1/M3)</b>		<b>R(M2/M4)</b>		<b>R(M4/M6)</b>	
<b>R(M1/M4)</b>		<b>R(M2/M5)</b>		<b>R(M4/M7)</b>	
<b>R(M1/M5)</b>		<b>R(M2/M6)</b>			
<b>R(M1/M6)</b>		<b>R(M2/M7)</b>		<b>R(M5/M6)</b>	
<b>R(M1/M7)</b>				<b>R(M5/M7)</b>	
		<b>R(M3/M4)</b>			
		<b>R(M3/M5)</b>		<b>R(M6/M7)</b>	
		<b>R(M3/M6)</b>			
		<b>R(M3/M7)</b>			



# Current Status

---

- Coding



- Final Design & Test

***TBD***

- Documentation

***TBD***



# Questions



“Well HAL, I’m damned if I can find anything wrong with it.”

“Yes. It’s puzzling! I don’t think I’ve ever seen anything quite like this before.”

-- *2001: A Space Odyssey*