Lab05
PRESENTATION

Chia Wei Kuo

Outline

- Strategies of Optimization
- Actions
- FSM

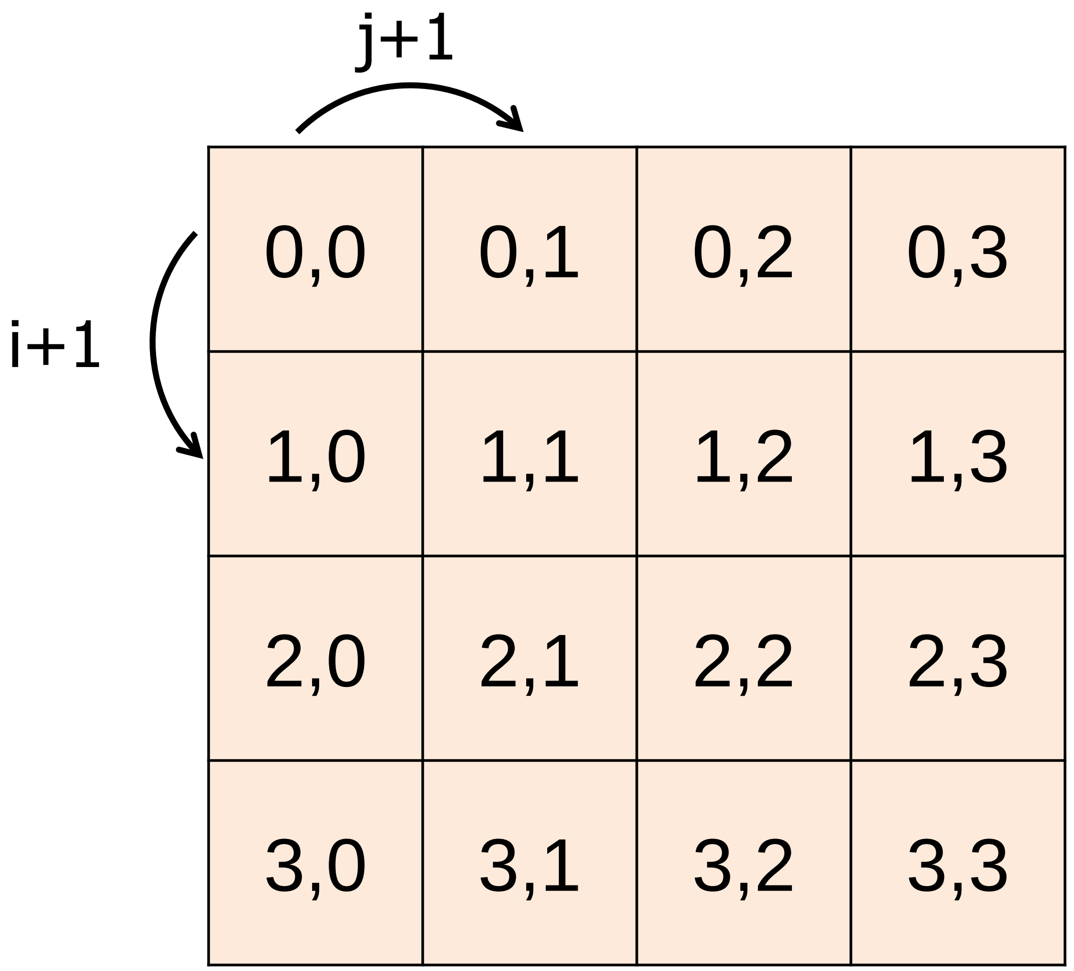
Strategies of Optimization

- Save area：
  - 1 SRAM 288 words
  - Sharing operators
  - Using registers as less as possible

- Save latency：
  - flip = change the way to read image.

Definition

- Initial image：
  - –(row,col) = (i,j)
  - –Address = {1’b0, i, j}



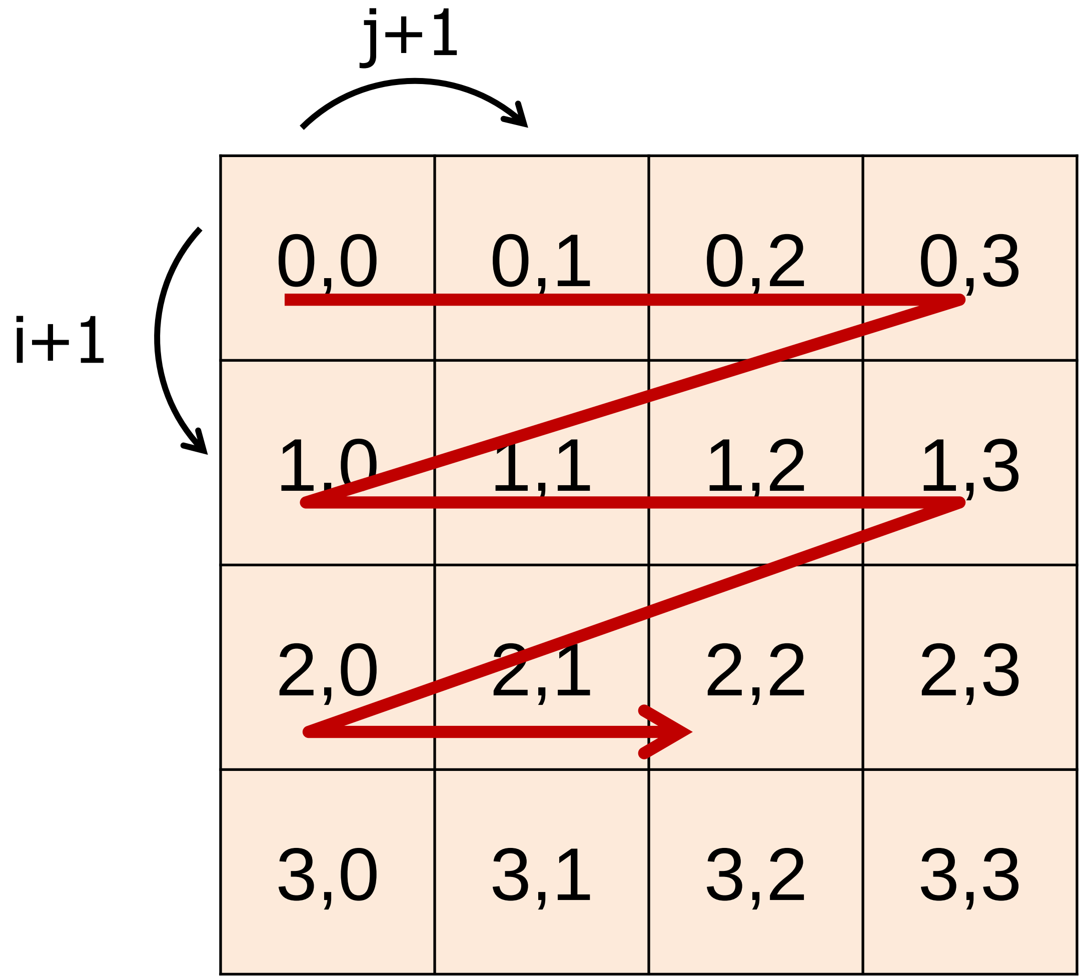
0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Flip

☹ Read & write memory => take a lot of time

☺ Change the way to read memory => 1 cycle

- Stored by register
  - (i_start,j_start)：start point (0,0)
  - j_first：j change first? 1
  - j_step：the value of j change +1
  - i_step：the value of i change +1



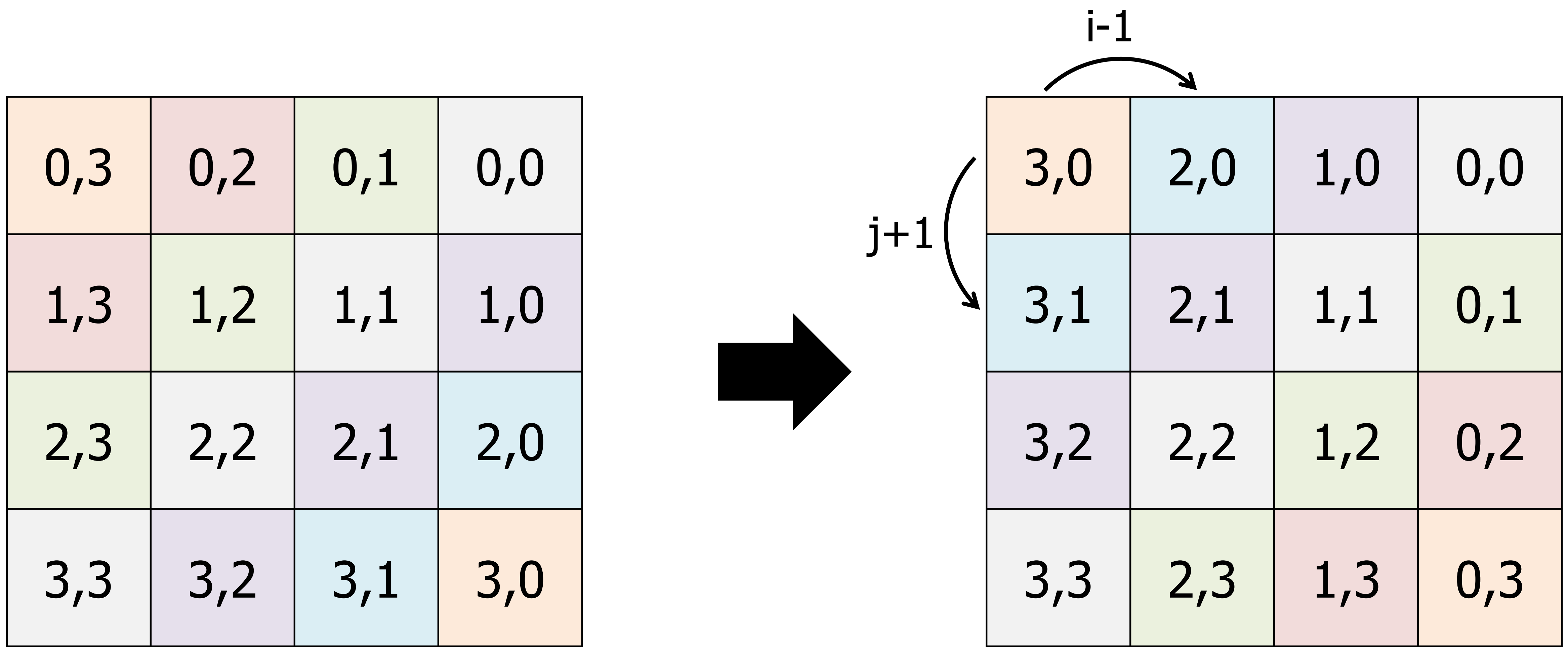
0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Table

Param. Flip	i_start	j_start	j_first	j_step	i_step
Horizontal	(j_first)? x:inverse	(j_first)? inverse:x	x	(j_first)? inverse:x	(j_first)? x:inverse
Vertical	(j_first)? inverse:x	(j_first)? x:inverse	x	(j_first)? x:inverse	(j_first)? inverse:x
Left-diagonal	inverse	inverse	inverse	inverse	inverse
Right-diagonal	x	x	inverse	x	x

x：don’t change inverse(expect j_first)：0<-> img_size-1 inverse(j_first)：0<->1

Example：Left-diagonal



0,3	0,2	0,1	0,0
1,3	1,2	1,1	1,0
2,3	2,2	2,1	2,0
3,3	3,2	3,1	3,0

Start point=(0,3)
j_first =1
j_step =-1
i_step =+1

3,0	2,0	1,0	0,0
3,1	2,1	1,1	0,1
3,2	2,2	1,2	0,2
3,3	2,3	1,3	0,3

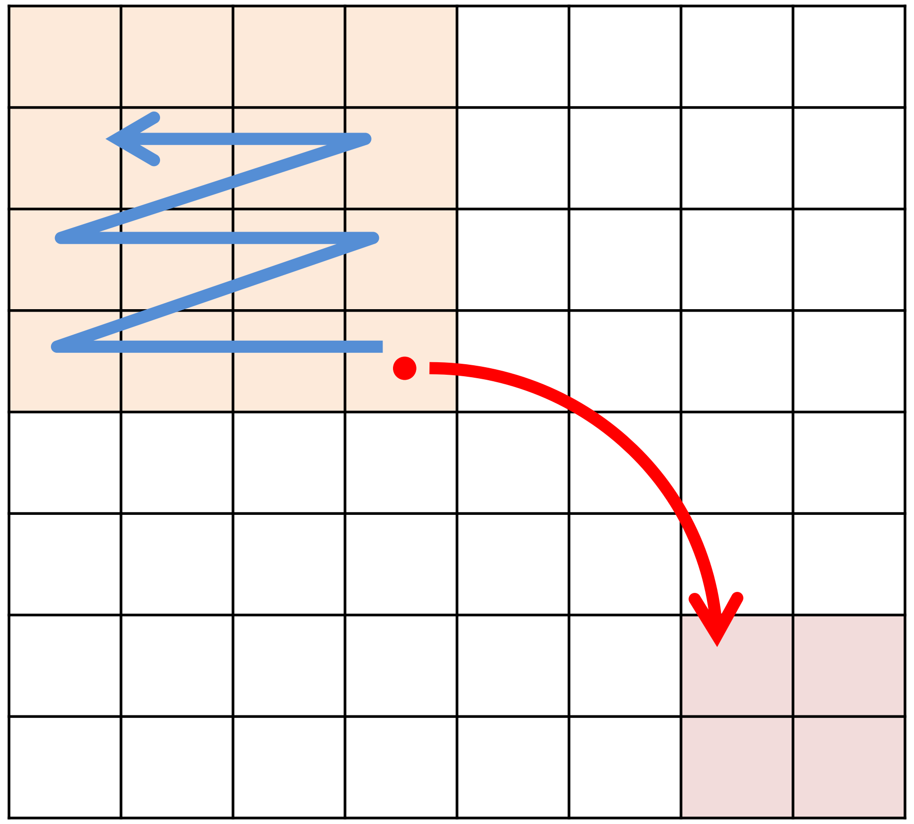
Start point=(3,0)
j_first =0
j_step =+1
i_step =-1

Just inverse all parameters!

Zoom-in

- Read from the last pixel to prevent overwriting the data we still need.
  - No matter what kinds of flips you do previously, this method is always worked. Just being care for the sequence of 4 write back data.

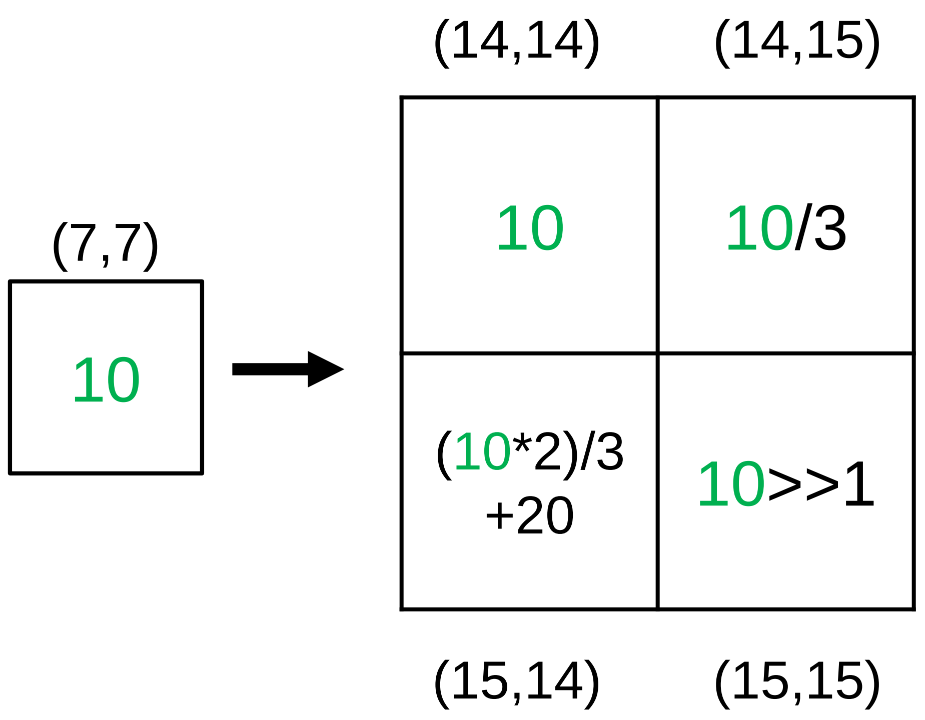
Image	Last pixel	Write Back to
4*4	(3,3)	(6,6) (6,7) (7,6) (7,7)
8*8	(7,7)	(14,14) (14,15) (15,14) (15,15)

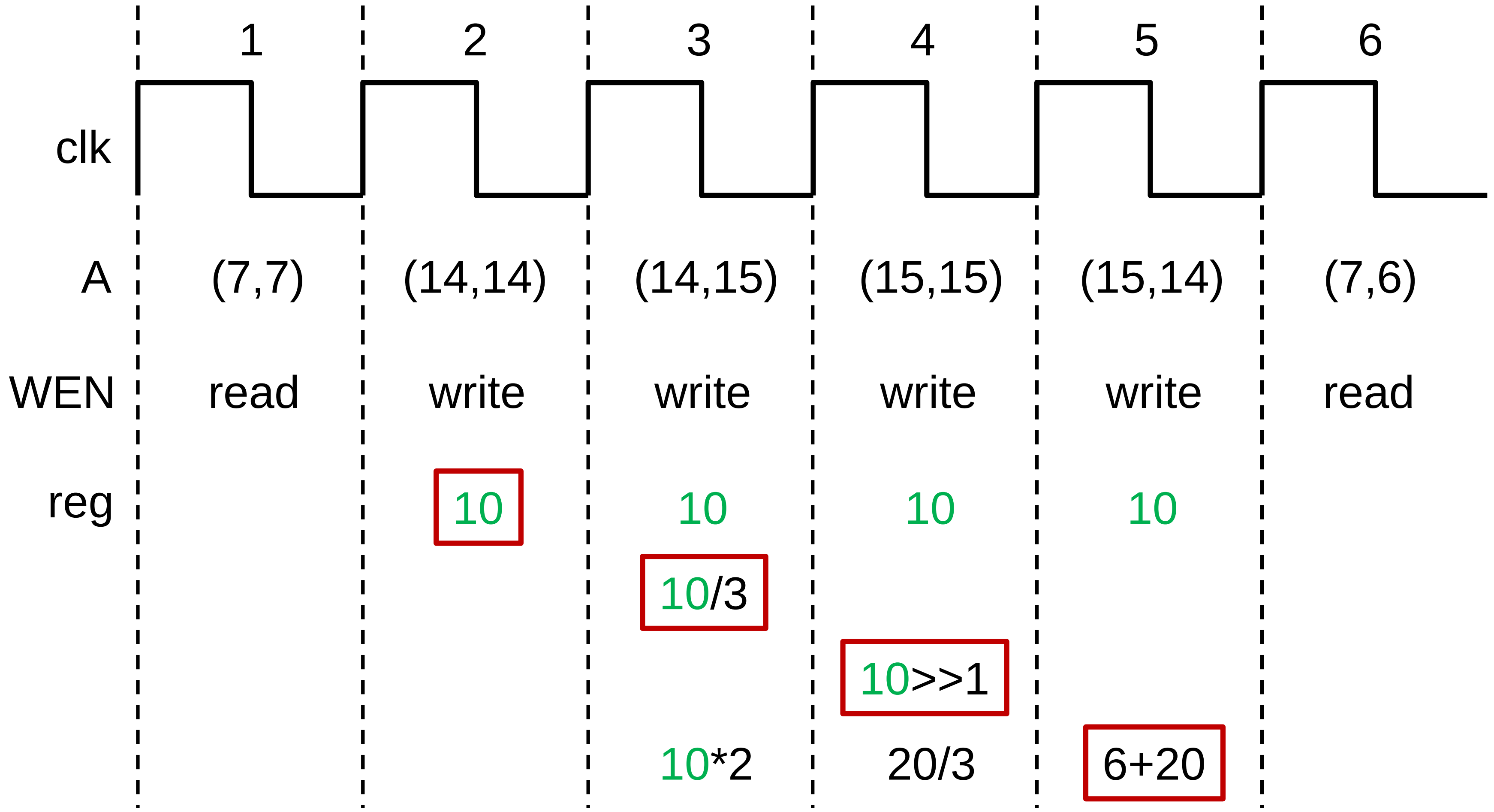


Zoom-in

- Reusing div by 3. (save area)
- Arranging when to calculate the result and write back.

Example：





	1	2	3	4	5	6
clk						
A	(7,7)	(14,14)	(14,15)	(15,15)	(15,14)	(7,6)
WEN	read	write	write	write	write	read
reg		10	10 10/3 10*2	10 10>>1 20/3	10 6+20	

Other Actions

- **Max Pooling、 Shortcut + Brightness Adjustment**
  - The sequence of pixels won’t affect the result.
  - Choosing the start point for easily defining where to write back.

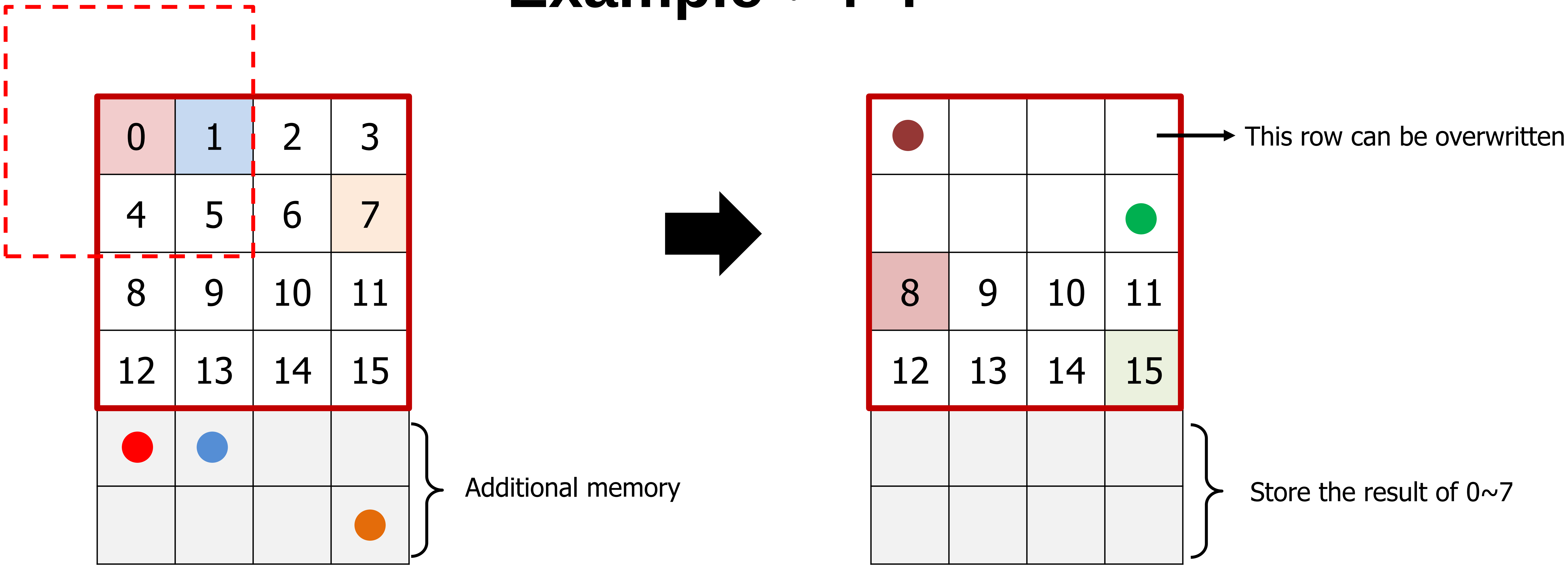
SB	4*4	8*8	16*16
Read start point	0,0	2,2	4,4
Write start point	0,0	0,0	0,0

Table. Shortcut + Brightness Adjustment

Cross Correlation

- Using 1 multiplier and 1 adder.
- Using additional memory to store a part of correlation results prevents the data which we still need being overwritten.

Example：4*4



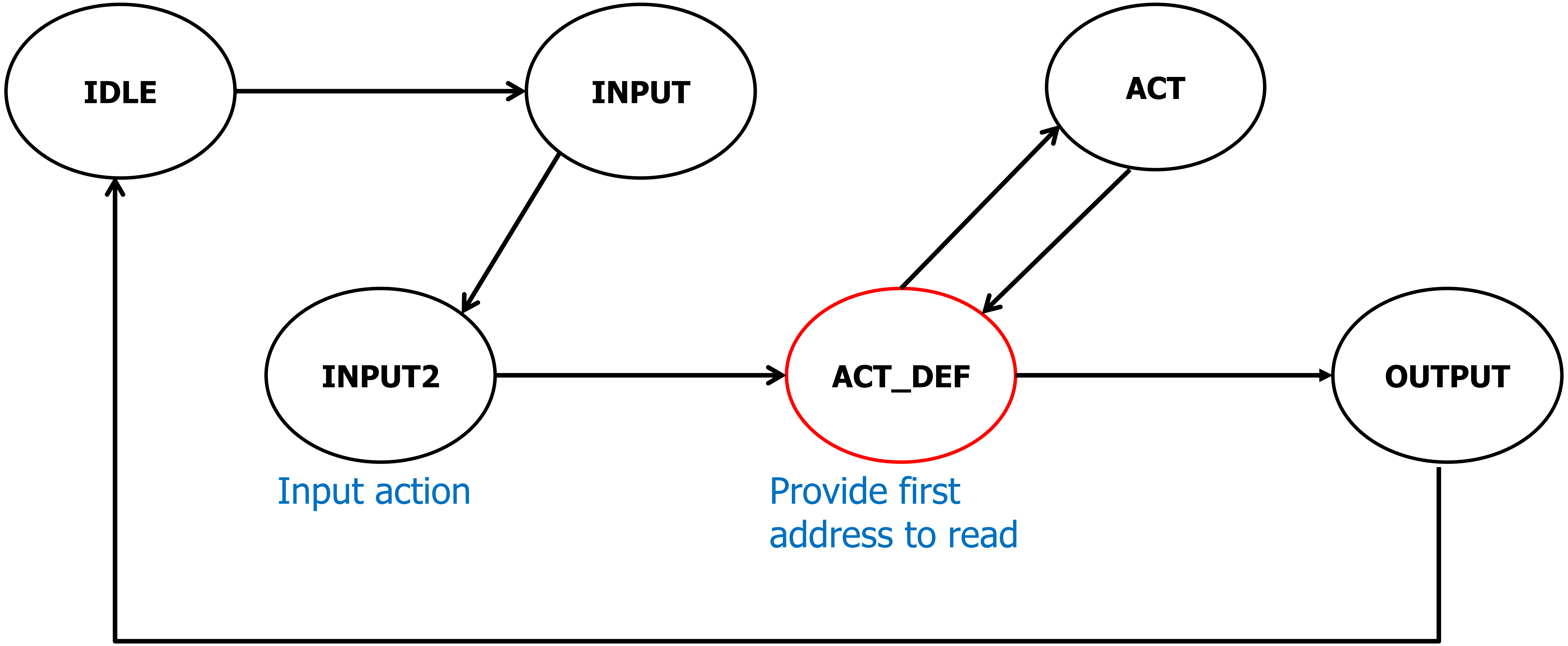
This row can be overwritten

Additional memory

Store the result of 0~7

» For 16*16 we need 256+2*16=288 to store the result.

FSM



Input image

7 actions

Input action

Provide first address to read

Thanks For Your Listening