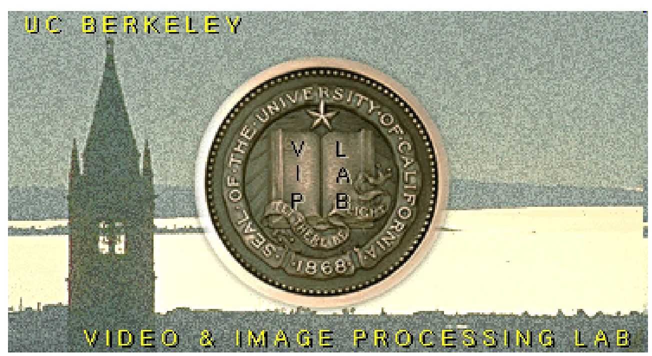
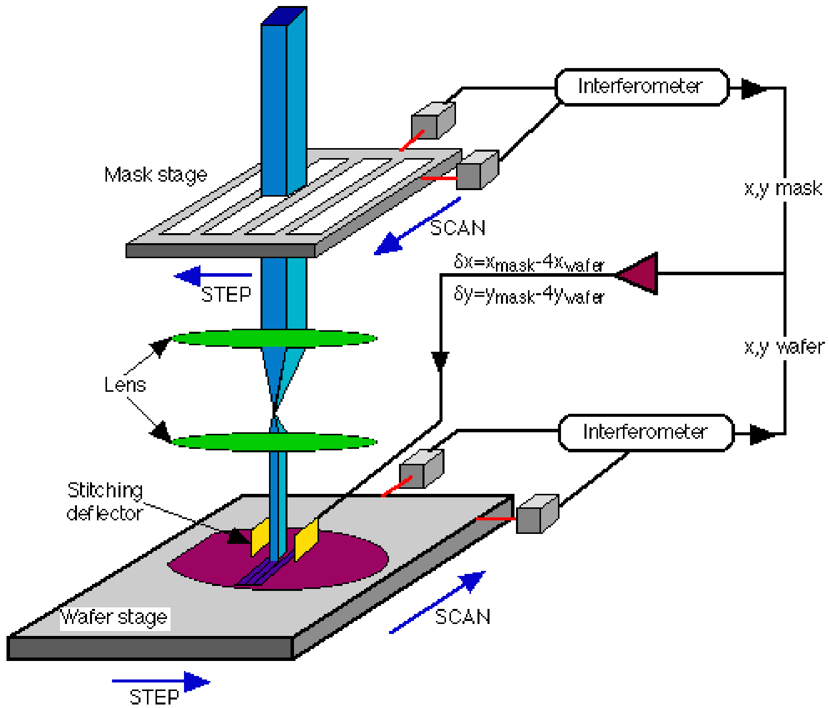
Lossless Compression Algorithms for Direct-Write Lithography Systems

Hsin-I Liu

Video and Image Processing Lab
Department of Electrical Engineering and Computer Science
University of California at Berkeley



Optical Lithography



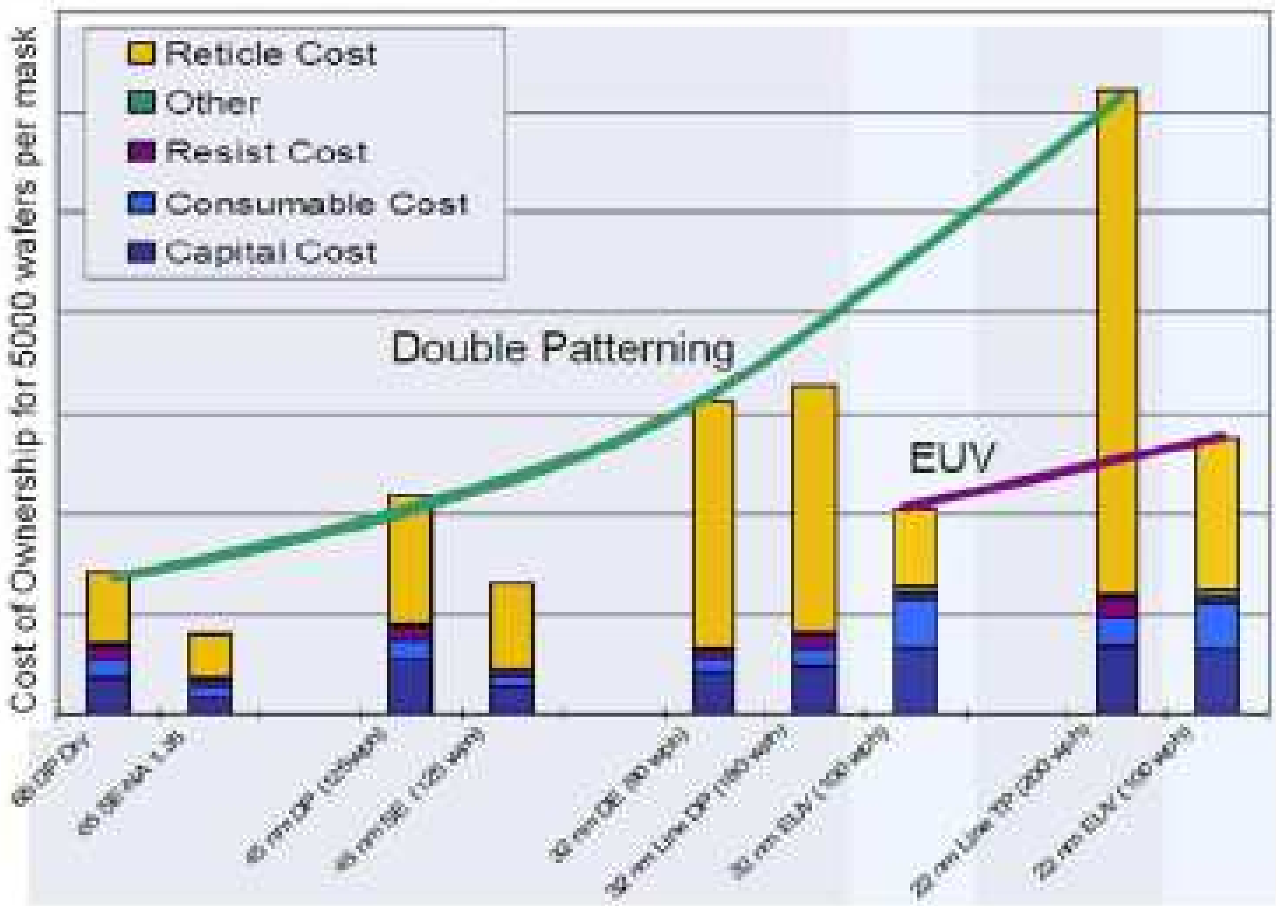
- Lithography is applied to create patterns on the wafer in semiconductor manufacturing
- Current approach: Mask is applied in optical lithography systems
  - cost of mask is increasing……

From Mask to Maskless Lithography

High Volumn Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Technology Node (nm)	90	65	45	32	22	16	11	8

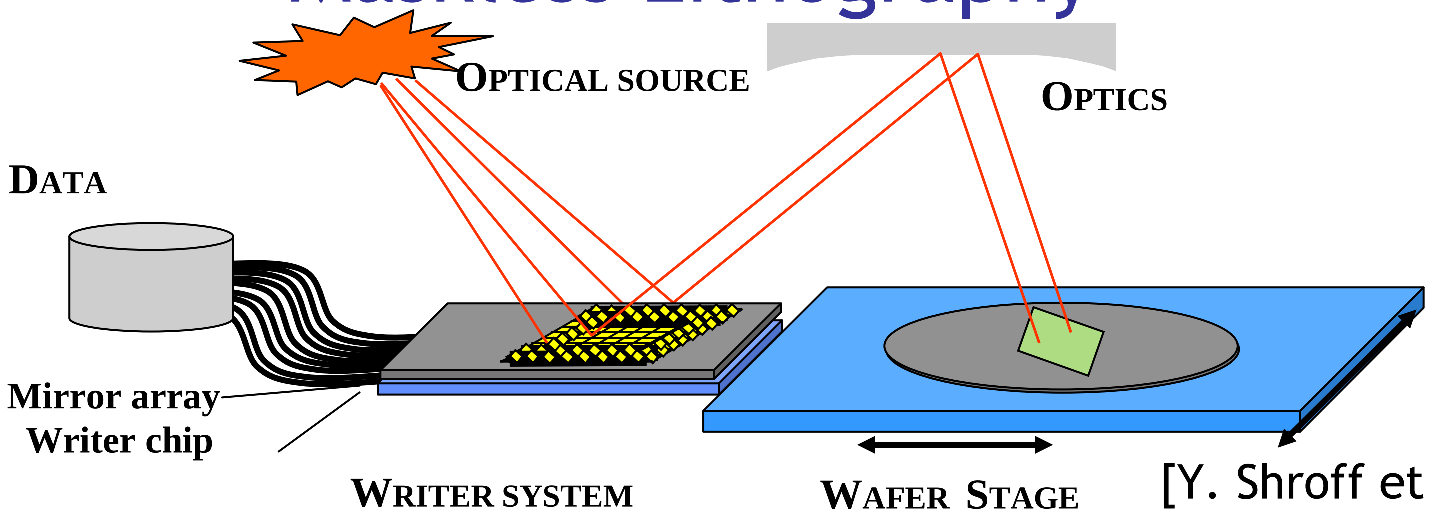
Source: ITRS 2004

Cost of Masks in Optical Lithography



ITRS 2009

Maskless Lithography

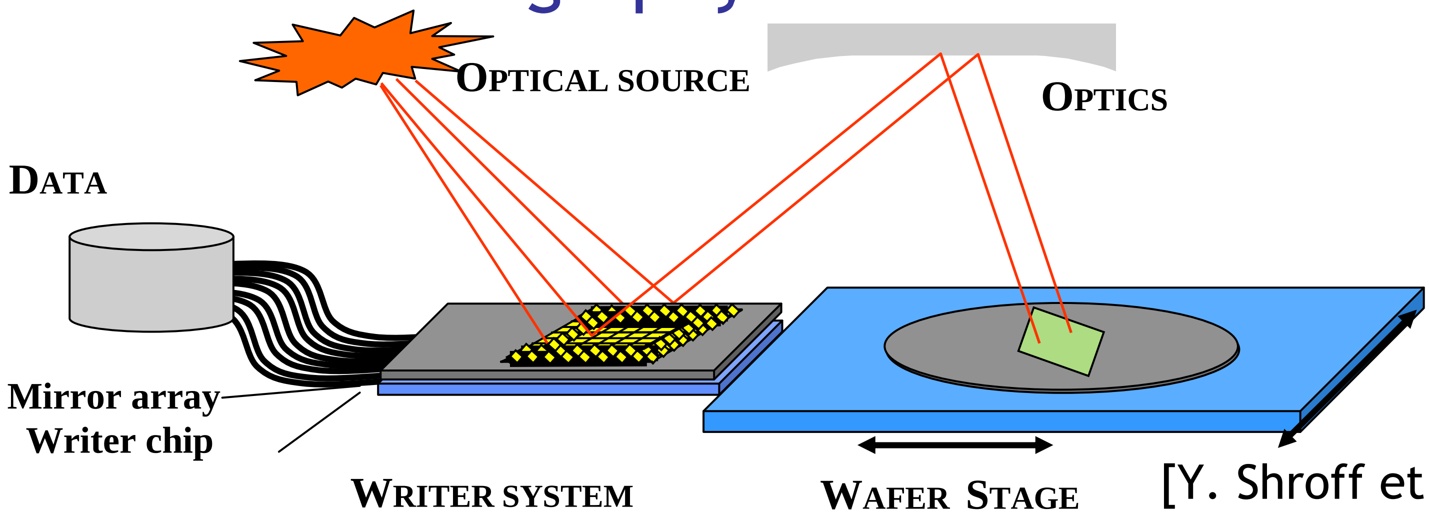


[Y. Shroff et al. ’00]

- A micromirror array is used to replace the optical mask
  - Reduce the cost of mask by x times
  - Increase patterning flexibility
- Focus of research:
  - Fabricate micromirror → array
  - Modify the layout pattern for proximity effect correction → OPC or EPC

However……

Maskless Lithography – Practical Issues



[Y. Shroff et al. '00]

- Each micromirror is controlled individually and dynamically
- Layout image is rasterized into pixel based

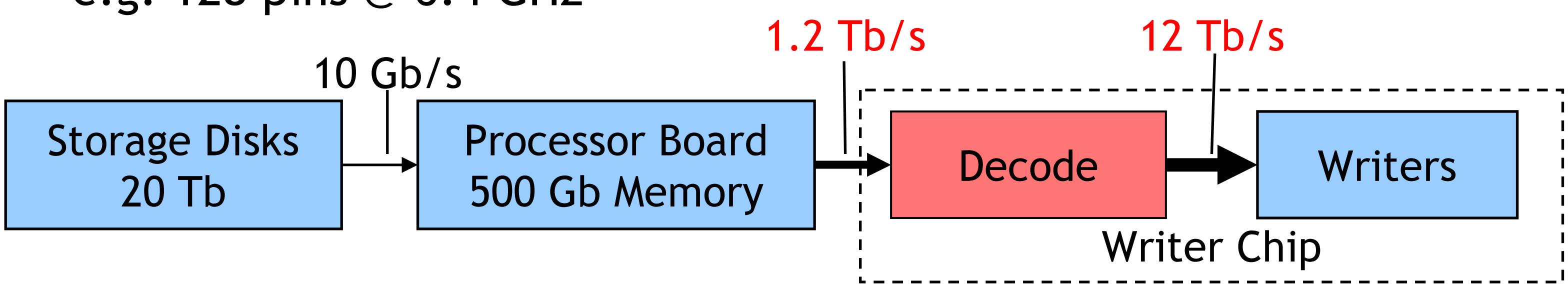
  → Data delivery problem for real-time manufacturing
- Update the pixel value for
  - Different portion of layout images
  - Overcome the voltage attenuation problem

Data Delivery Issue

- Data rate for 45nm minimum feature to achieve 1 wafer layer/minute throughput

$$\frac{wafer \cdot layer}{60\ s} \times \frac{\pi/4 \cdot (300\ mm)^2}{wafer \cdot layer} \times \frac{pixel}{(22\ nm)^2} \times \frac{5\ bits}{pixel} = 12\ Tb/s$$

- Estimated needed compression: 12 Tb/s ÷ 1.2 Tb/s = 10
- Board to chip communication: 1.2 Tb/s
  - e.g. 128 pins @ 6.4 GHz



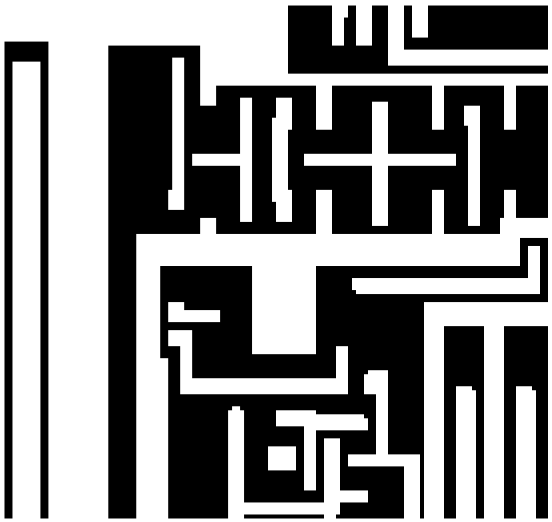
- Throughput requirement can be reduced to 3-5 wafer layers per hour → still need compression
- Lossless compression is applied to
  - Reduce storage space
  - Lower I/O throughput overhead

Data Compression Requirements

- Lossless compression
- Achieve ~10 compression efficiency
- Asymmetric compression algorithms
  - Offline encoding
  - Real-time decoding → decoder is implemented in hardware and integrated into the writer system

Block GC3 - Compression Algorithm for Rasterized, Flattened Layout

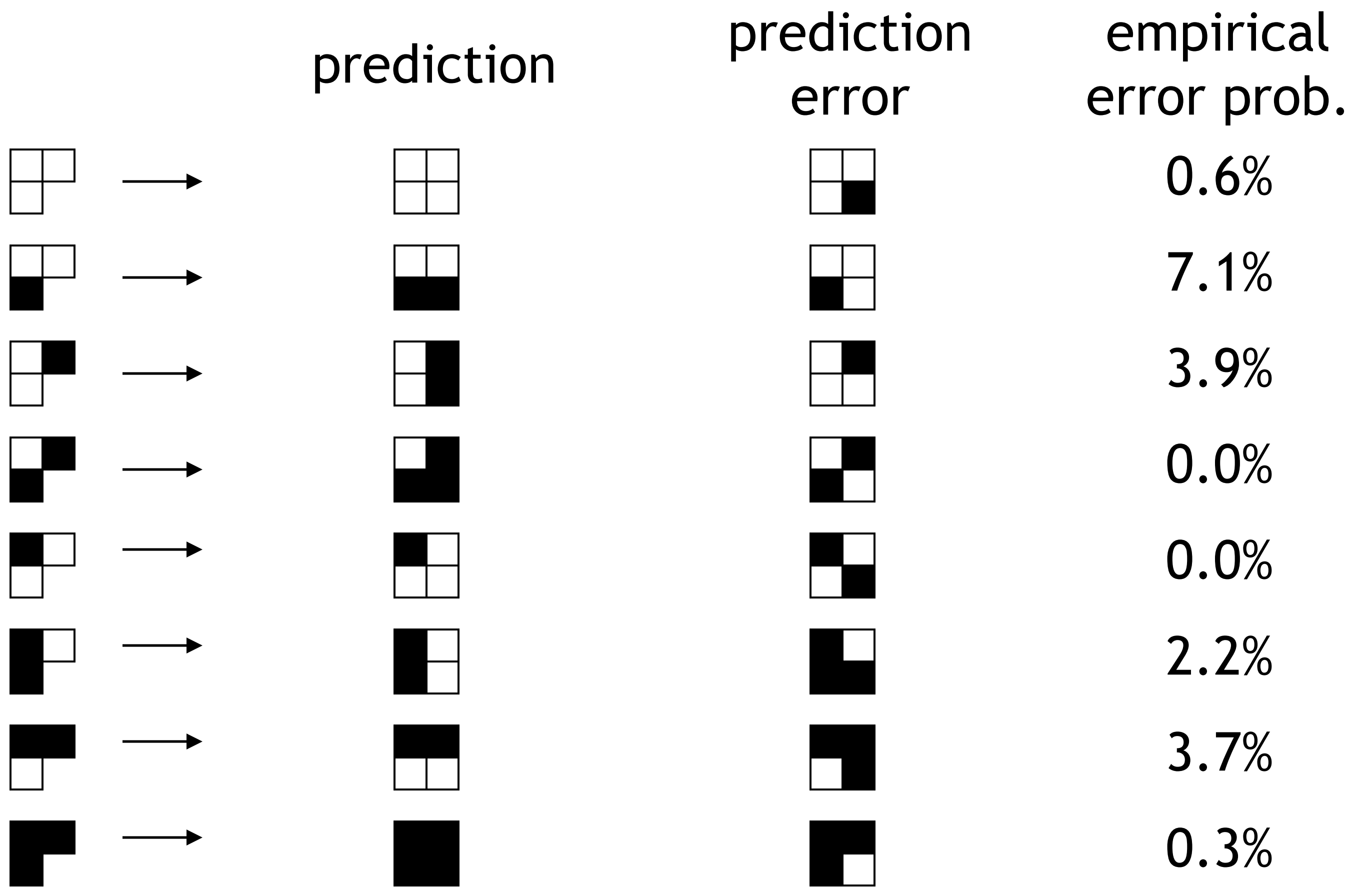
Block Golomb context copy code (Block GC3)



- Prediction from Context - JBIG
  1. Predict a pixel value from neighboring pixels (P)
  2. Good for non-repetitive layouts

[H. Liu '06]

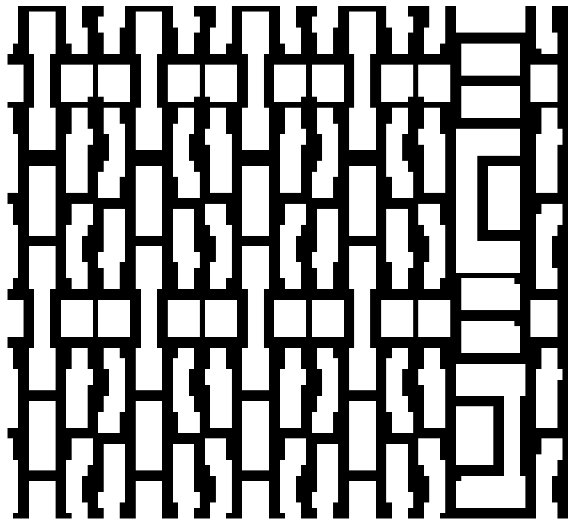
Block GC3 - Context Predict



a	*b*
c	*z*

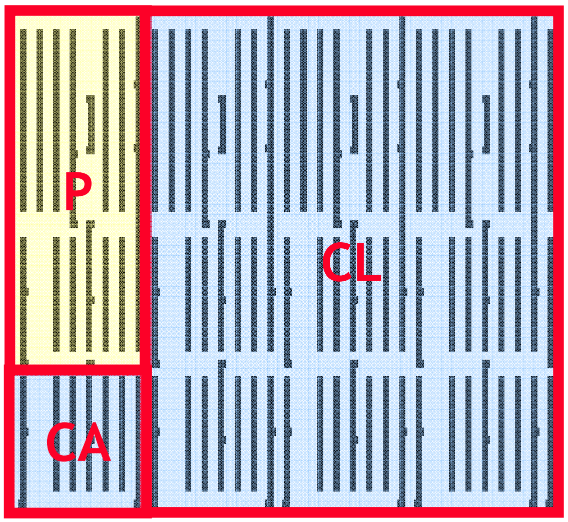
x = b - a + c
if (x < 0) then z = 0
if (x > max) then z = max
otherwise z = x

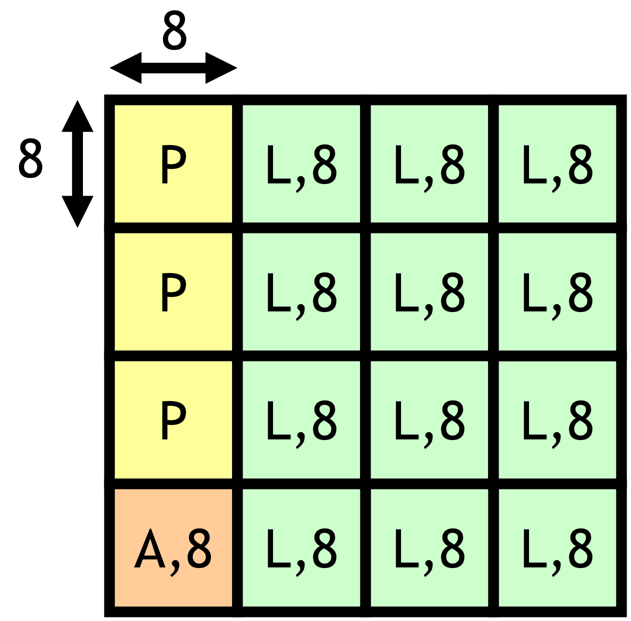
Block GC3 - Copy



- Copying – ZIP, 2D-LZ
  1. Copy from left or above
  2. Good for repetitive layouts

Block GC3 - Segmentation

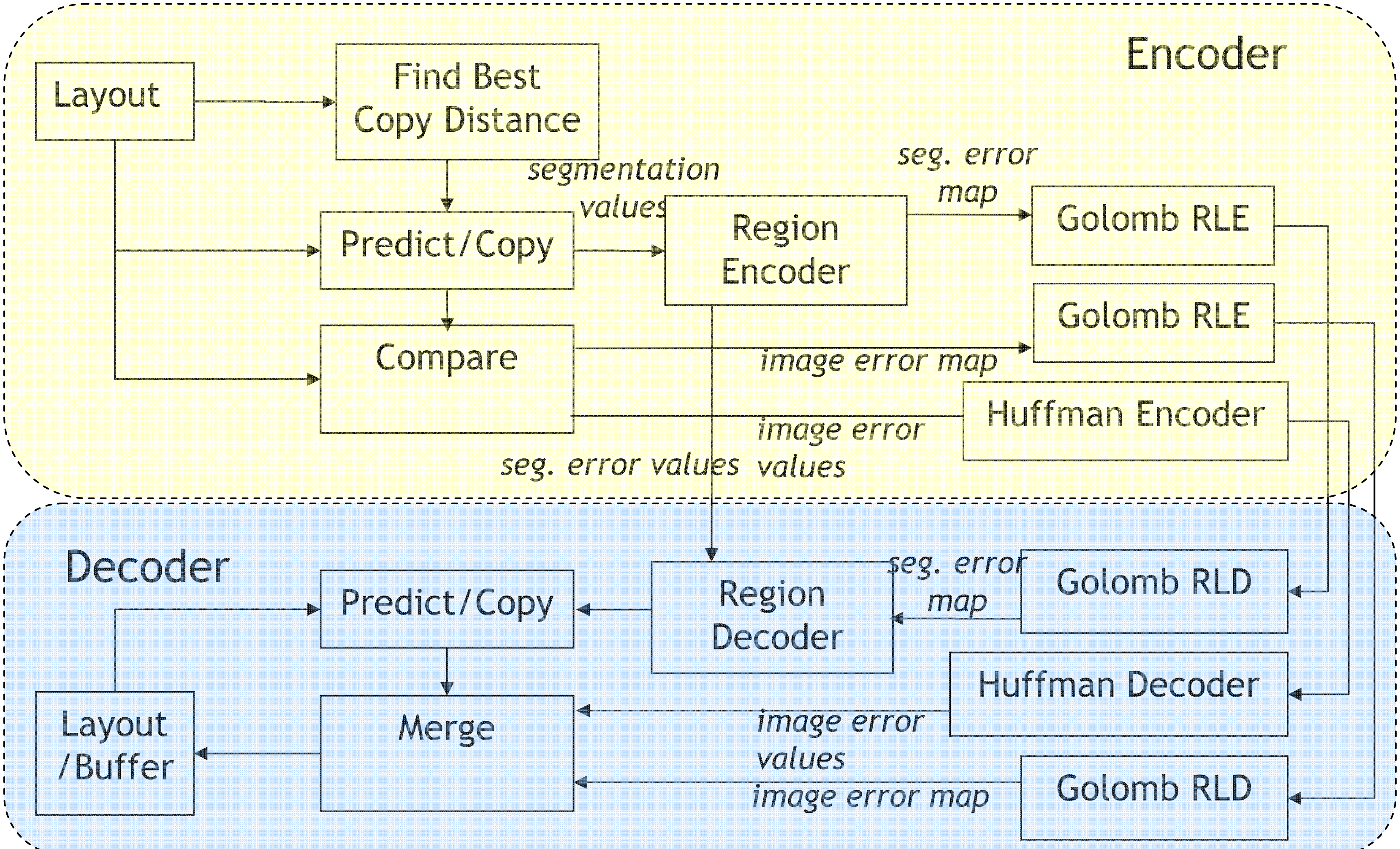




Block GC3 Segmentation map

- Layout images are divided into prediction and copy regions
  - Determined within 8 x 8 block
- Errors from prediction and copy are transmitted from Encoder to decoder
- All the information is further compressed

Block GC3 Encoder/Decoder Architecture



- Outperform the existing techniques
- Simple decoder design

[V. Dai '05]

Golomb Run-Length Code

- A simple code for binary stream

0001 0000 0000 001 1 001 01 .....

Bucket Size (B): maximum # of zeroes in a row

B = 4

Two kind of codes: (0) → B zeros in a row

(1, n) → n zeros in a row followed by a one

(1,3) (0) (0) (1,2) (1,0) (1,2) (1,1)

Compression achieved

Additional information introduced

Golomb Run-Length Code

- A simple code for binary stream

0001 0000 0000 001 1 001 01 .....

Bucket Size (B): maximum # of zeroes in a row

B = 4

Two kind of codes: (0) → B zeros in a row

(1, n) → n zeros in a row followed by a one

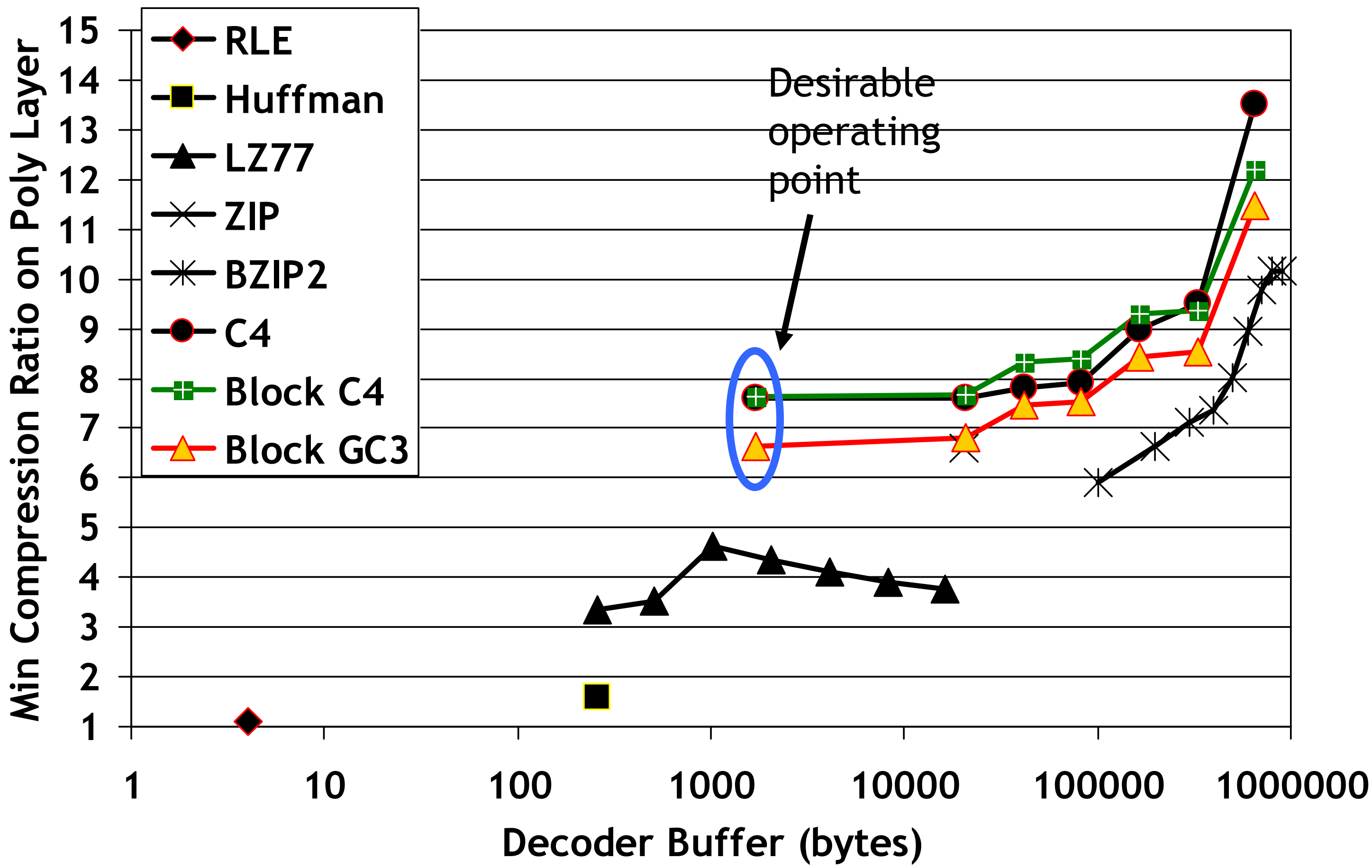
(1,3) (0) (0) (1,2) (1,0) (1,2) (1,1)

Golomb code achieves its best compression efficiency in i.i.d. random variables

→ achieves inefficient compression with highly skewed bitstream such as error location

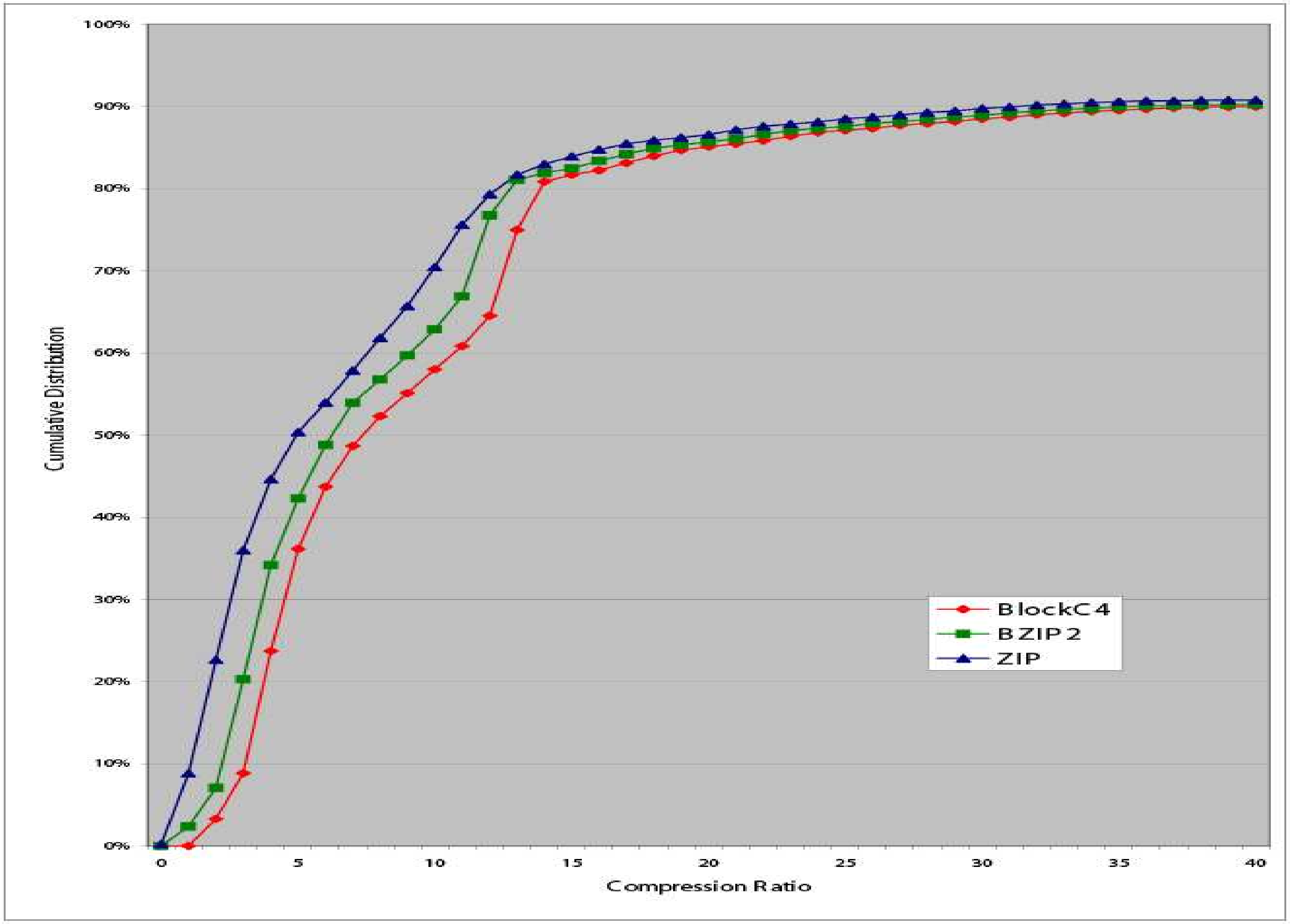
simple decoder design

Complexity vs. Compression Ratio of Compression Schemes



[H. Liu '06]

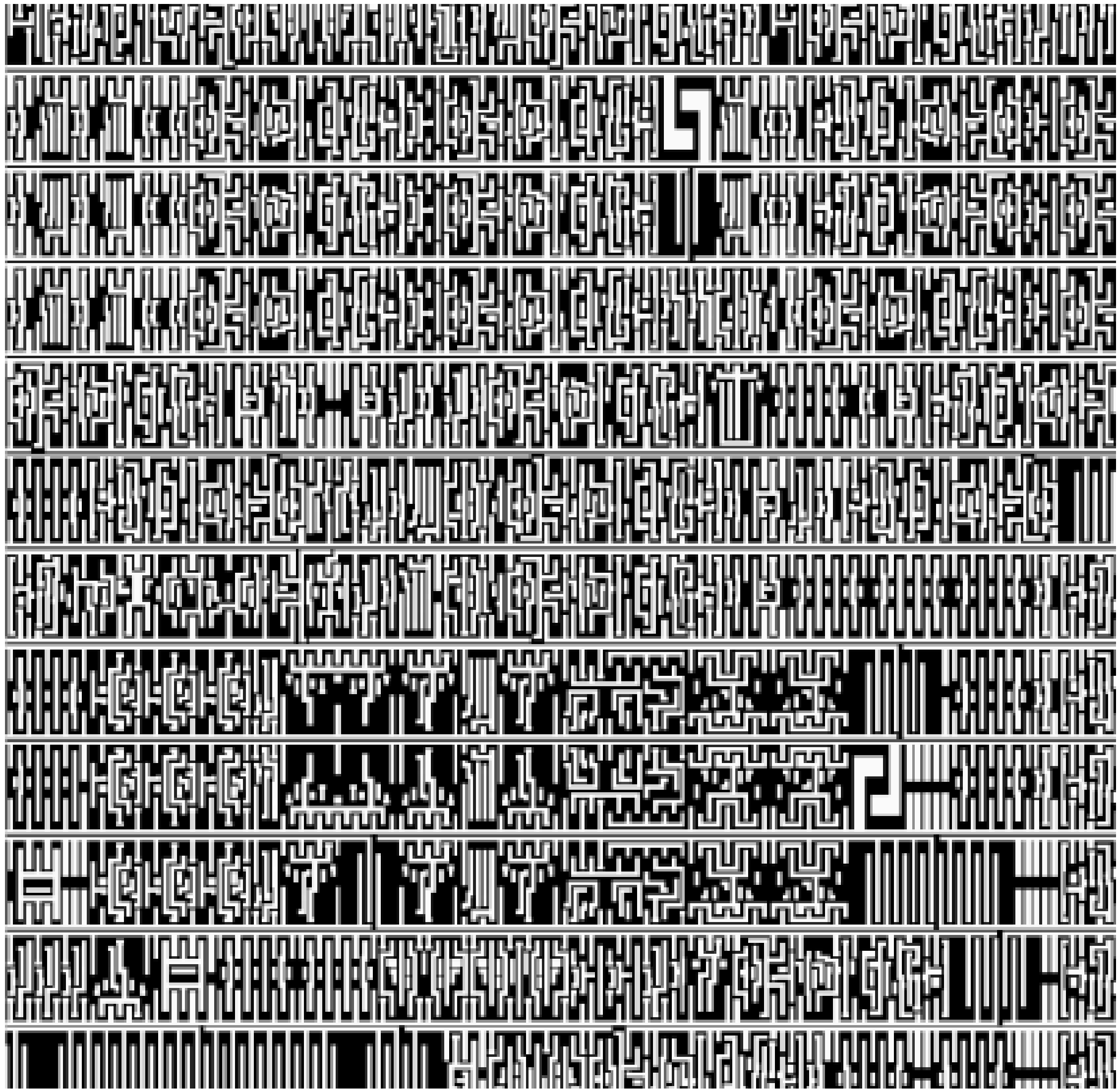
Full-Chip Test



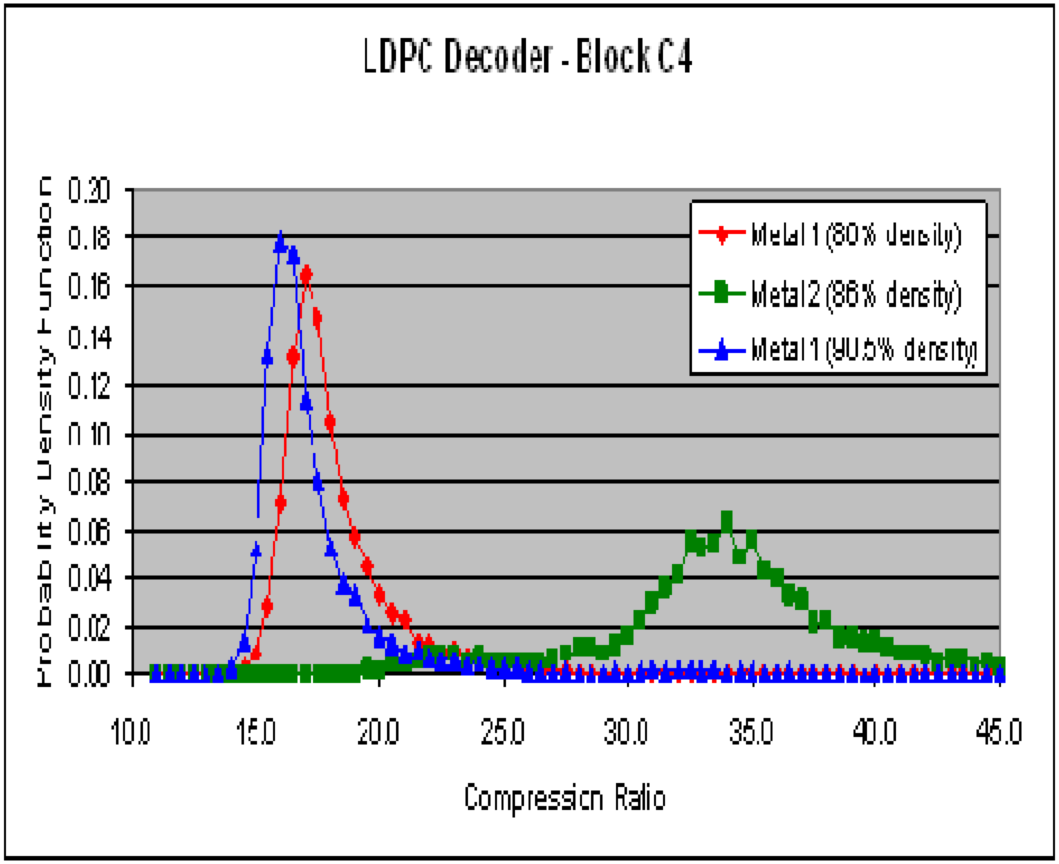
AMD CPU 65 nm Metal-1

- 24% of the images have CR < 5

[A. Zakhor '09]



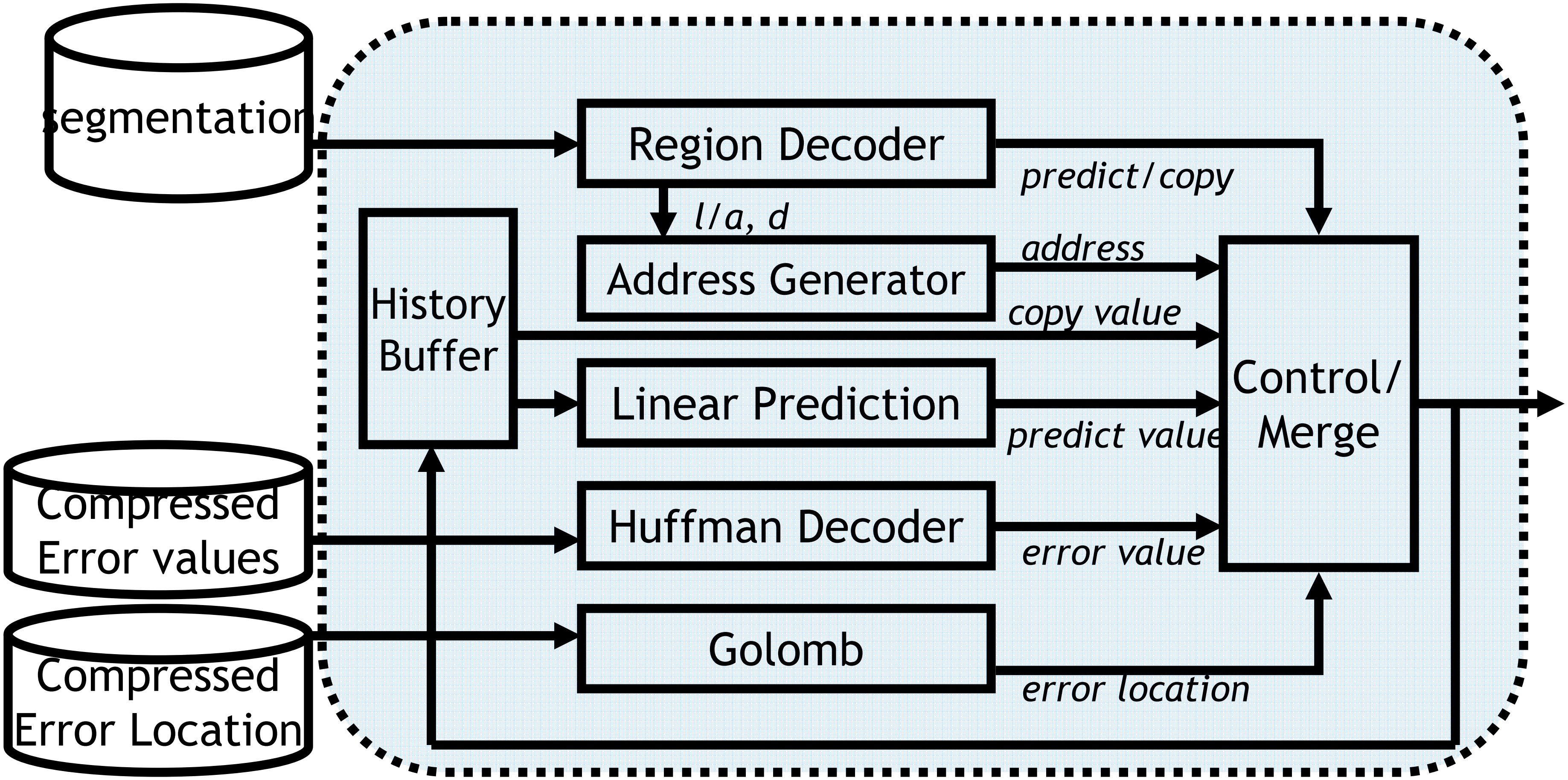
Full-Chip Test



ST ASIC 65 nm

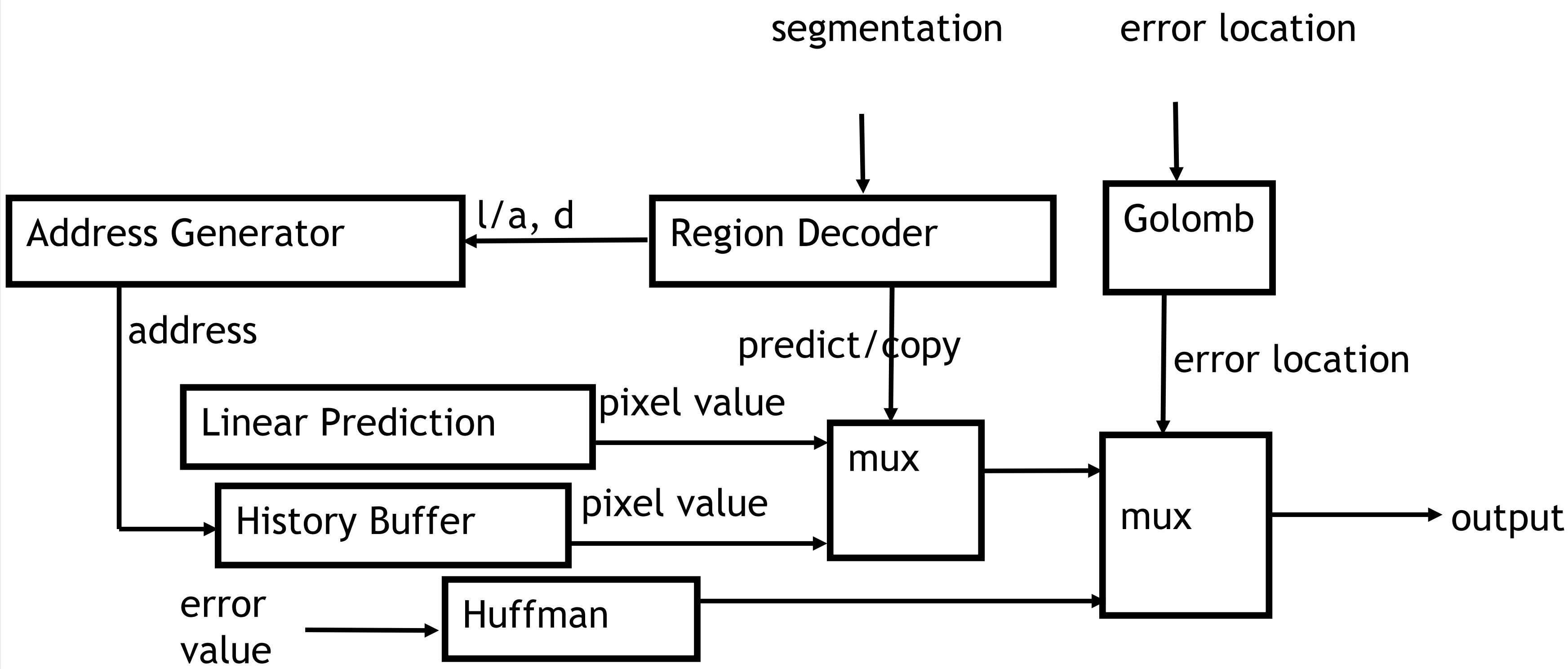
[A. Zakhor '09]

Block Diagram of Block GC3 Decoder

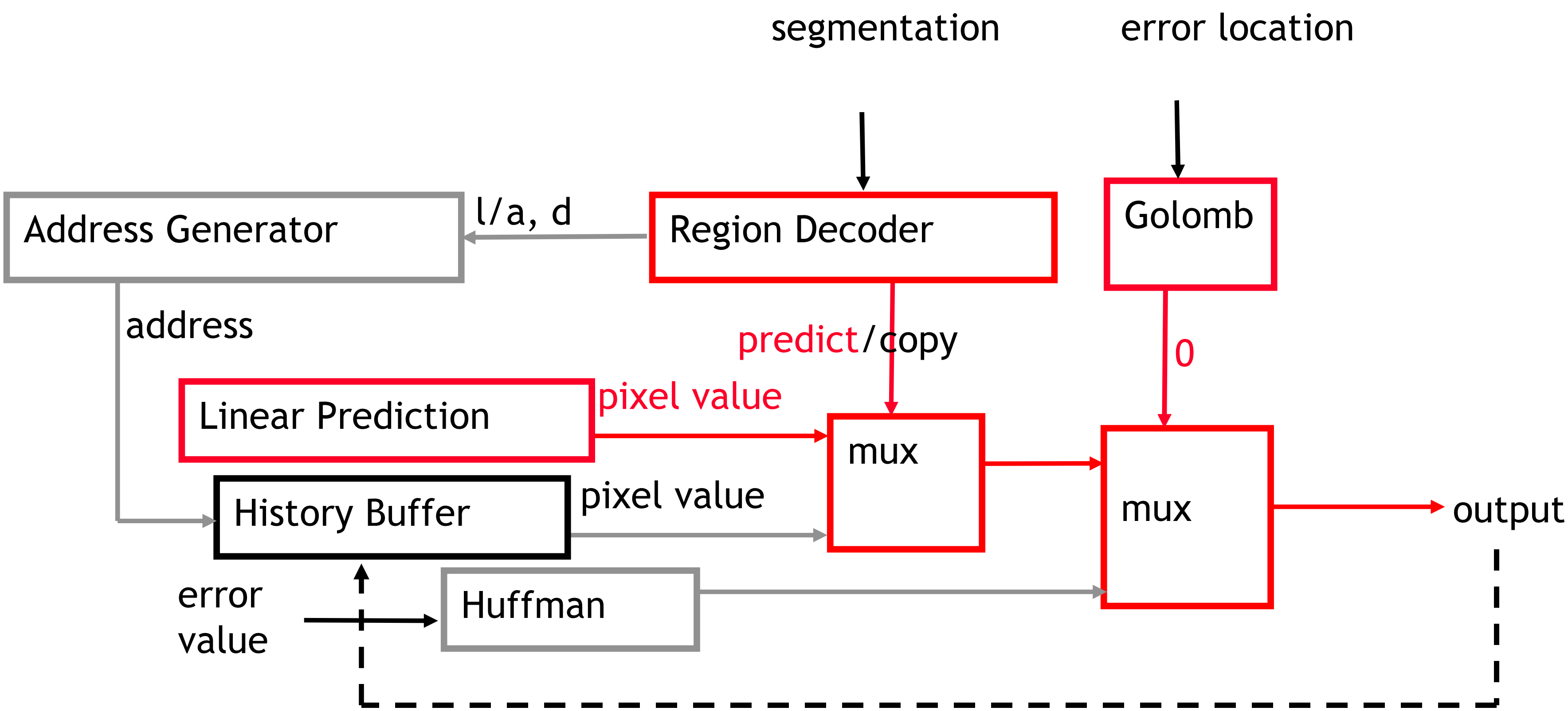


•High parallelism for hardware implementation→ Data flow architecture

Data Flow of Decoder

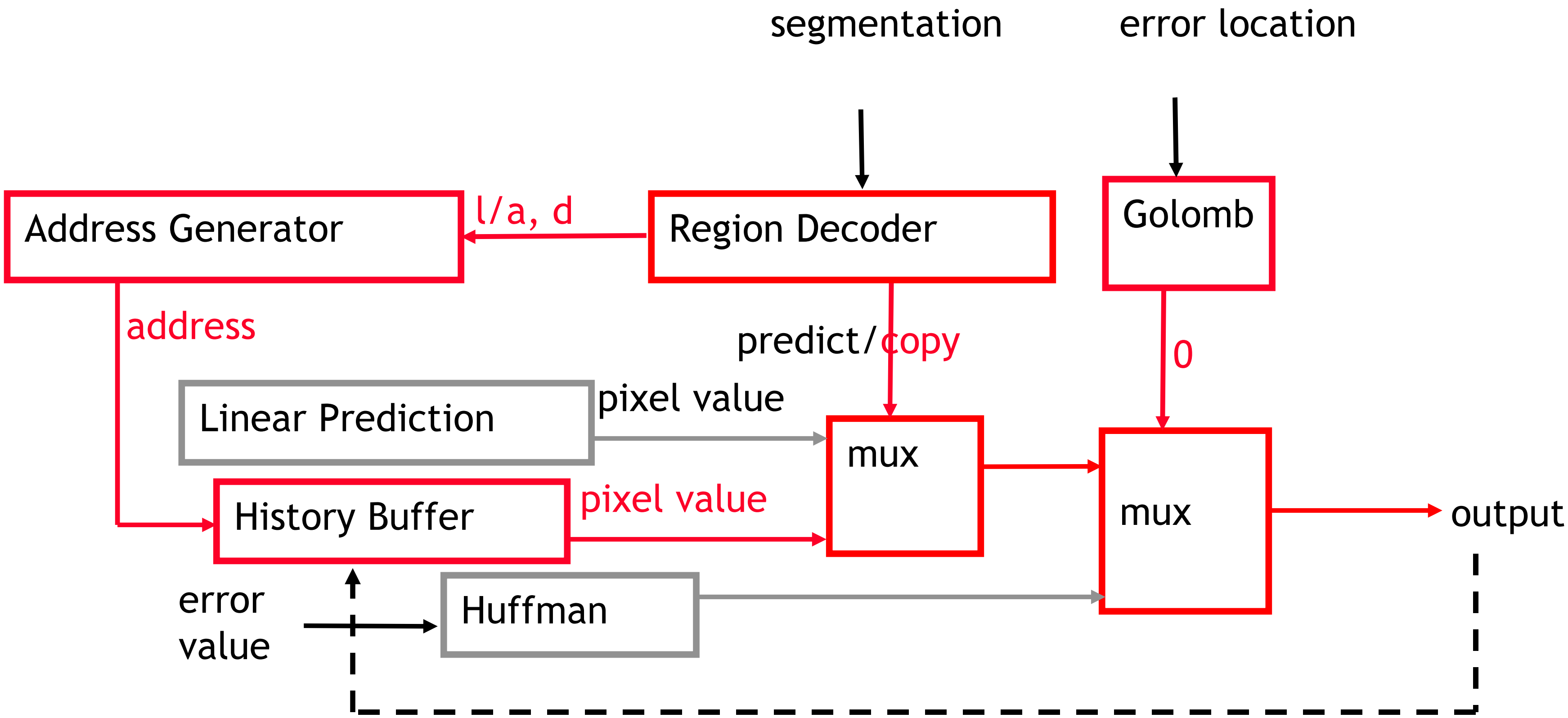


Data Flow of Decoder - Predict



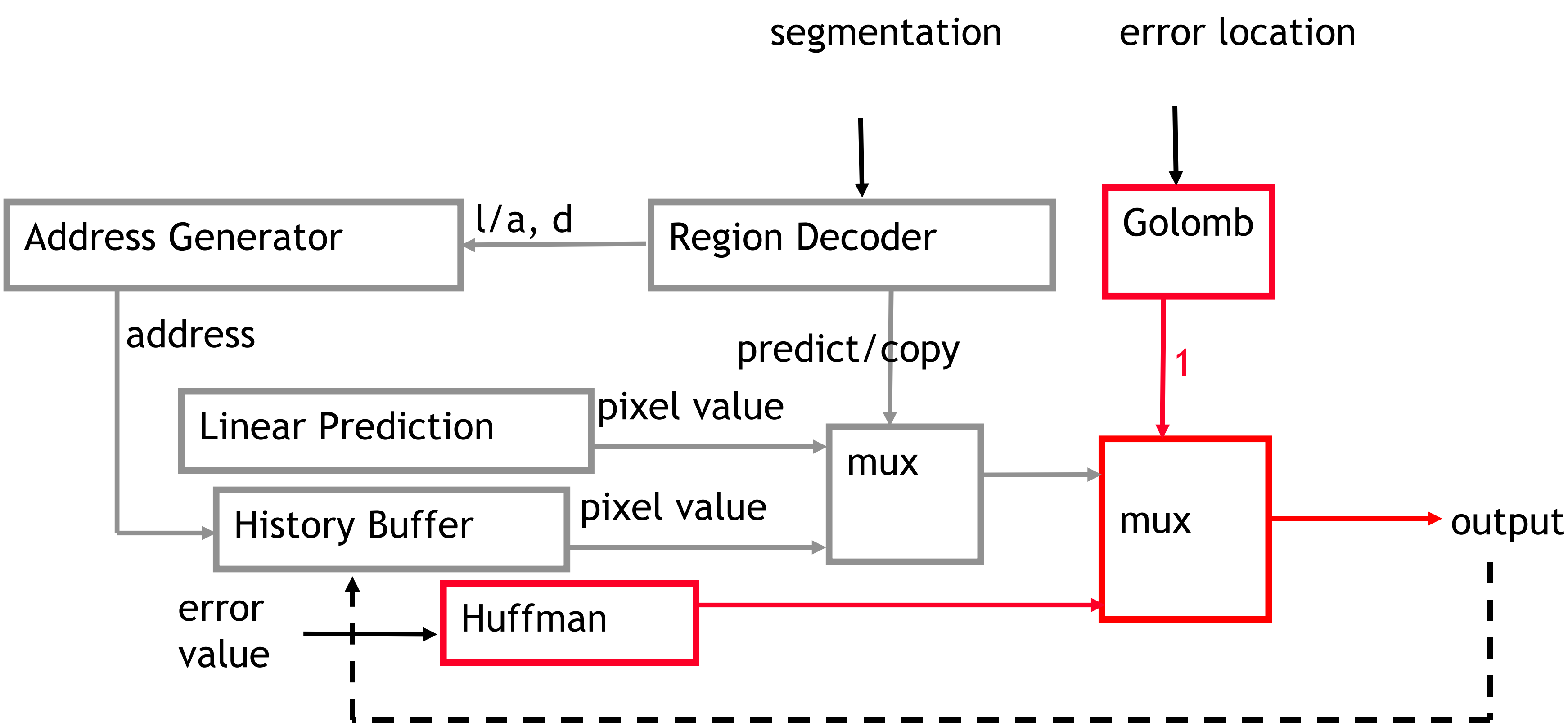
- After the decoding, the pixel value is stored back to history buffer

Data Flow of Decoder - Copy



- After the decoding, the pixel value is stored back to history buffer

Data Flow of Decoder - Error



- After the decoding, the pixel value is stored back to history buffer

Decoder Performance - FPGA

Device	Xilinx Virtex II Pro 70
Number of slice flip-flops	3,233 (4%)
Number of 4 input LUTs	3,086 (4%)
Number of block RAMs	36 (10%)
System clock rate	100 MHz
System throughput rate	0.99 (pixels/clock cycle)
System output data rate	495 Mb/s

- The hardware performance can be improved
  - Update FPGA devices
  - Apply ASIC implementation

Decoder Performance - ASIC

Block		Area (um^2)	Throughput (output/cycle)	Power (mW)
Golomb		1,136	1	0.2
Huffman		848	1/codeword+2	0.21
Linear Prediction		455	1	0.16
Address Generator		362	0.99	0.03
Region Decoder		18,370	1	7.26
Control/Merge		749	1	0.22
Memory		46,960	1	13.27
Block GC3	Single decoder	69,288	0.99	21.48

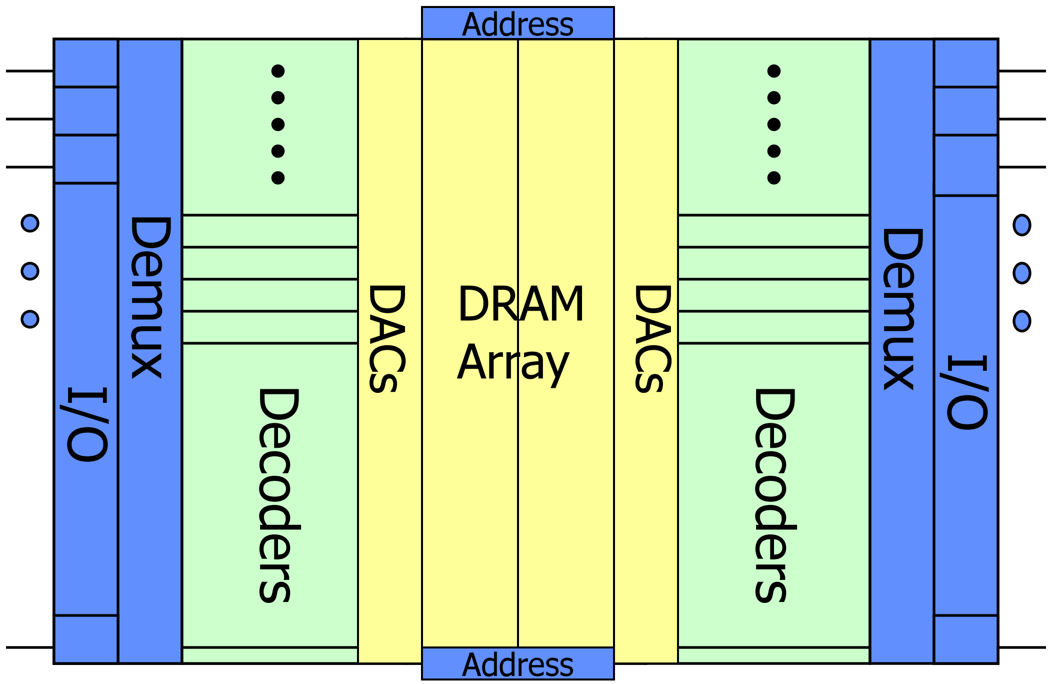
- 85% of area results from 1.7 KB of memory
- System clock rate: up to 500 MHz
- System throughput: 0.99
- System output rate: up to 2.47 Gb/s
- 200 decoders to achieve 500 Gb/s → 3 wafer layers per hour

Apply Block GC3 to reduce I/O overhead

I/O Type	Data rate	# of link for 500 Gb/s	# of link with Block GC3
Cell I/O	6.4 Gb/s	80	12
Hyper Transport 3.1	6.4 Gb/s	80	12
Optical link	3 Gb/s	167	26
Intel 65 nm interface	10 Gb/s	50	8
Intel 45 nm interface	25 Gb/s	20	3

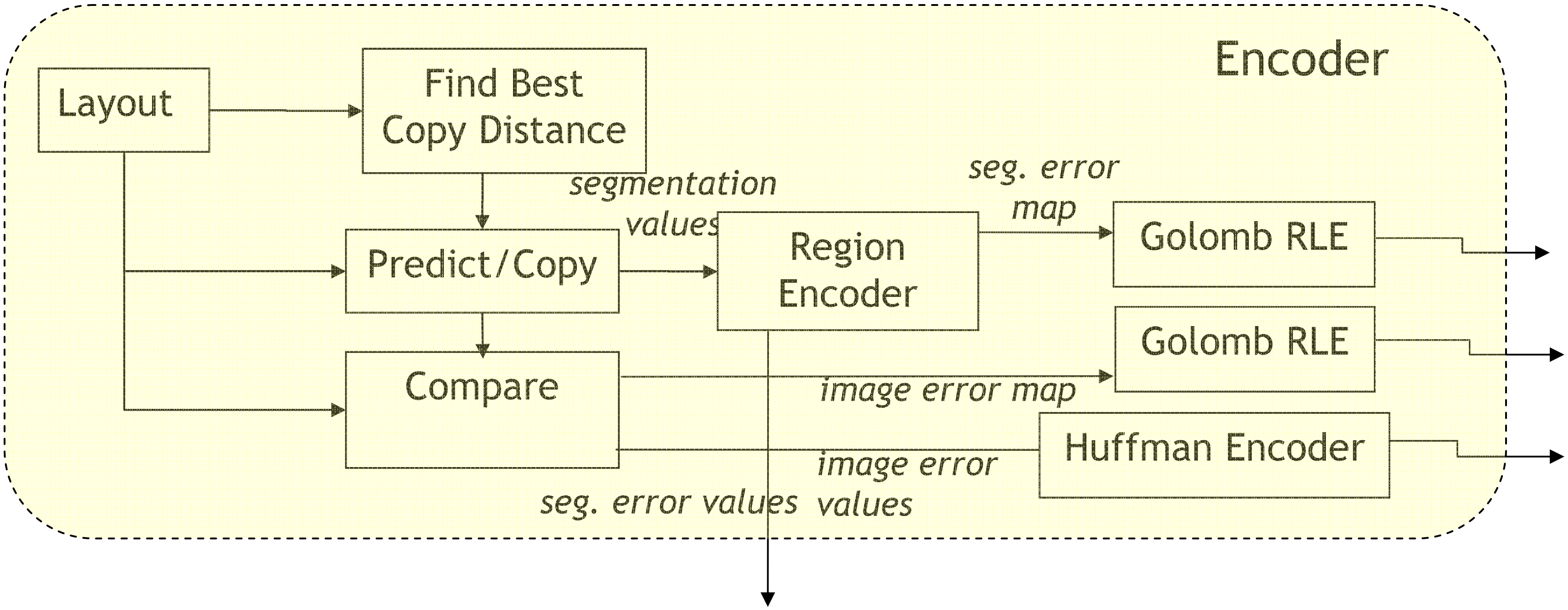
- 200 Block GC3 decoders is 14 mm^2
- Reduced I/O interface is more practical for direct-write applications

Writer Chip Architecture



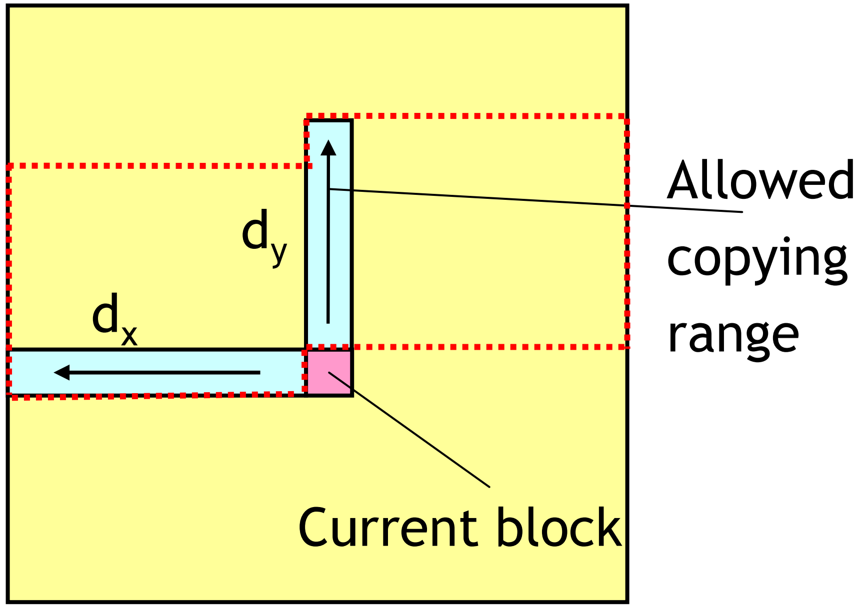
- DRAM array directly controls the micromirror array above
- Throughput of the chip: 3 wafer·layer/hour (500Gb/s)

Encoding complexity of Block GC3



- Find best copy distance → the most computational challenging part of encoding

Find the Best Copy Distance



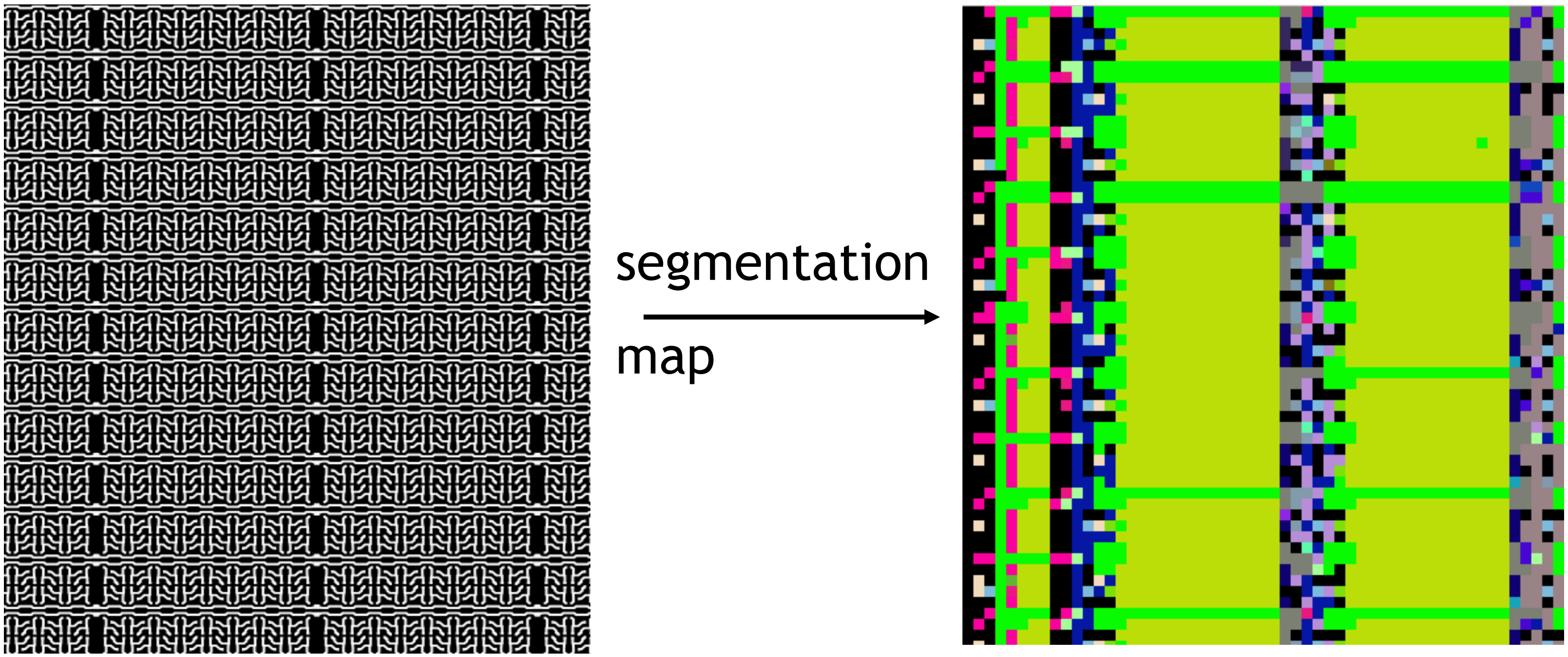
- For an m x n image with block size M, the complexity is

$$O\left[\frac{mn}{M^2}(d_x + d_y)\right]$$

Memory size= d_x x d_y

- Block segmentation reduces the complexity by M^2
- For linear writing system, horizontal/vertical copy is sufficient

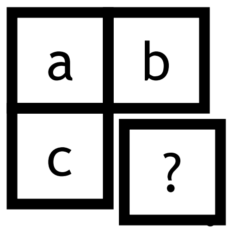
Find the Best Copy Distance – Multiple Candidates



- Every block may have more than one candidates with fewest mismatches → enforce spatial coherency for better compression
- Region growing → use the fewest number of regions to represent the segmentation map

Region Growing

- 2-D region growing is an NP-complete problem
  - Use left/above segmentation info as preferences



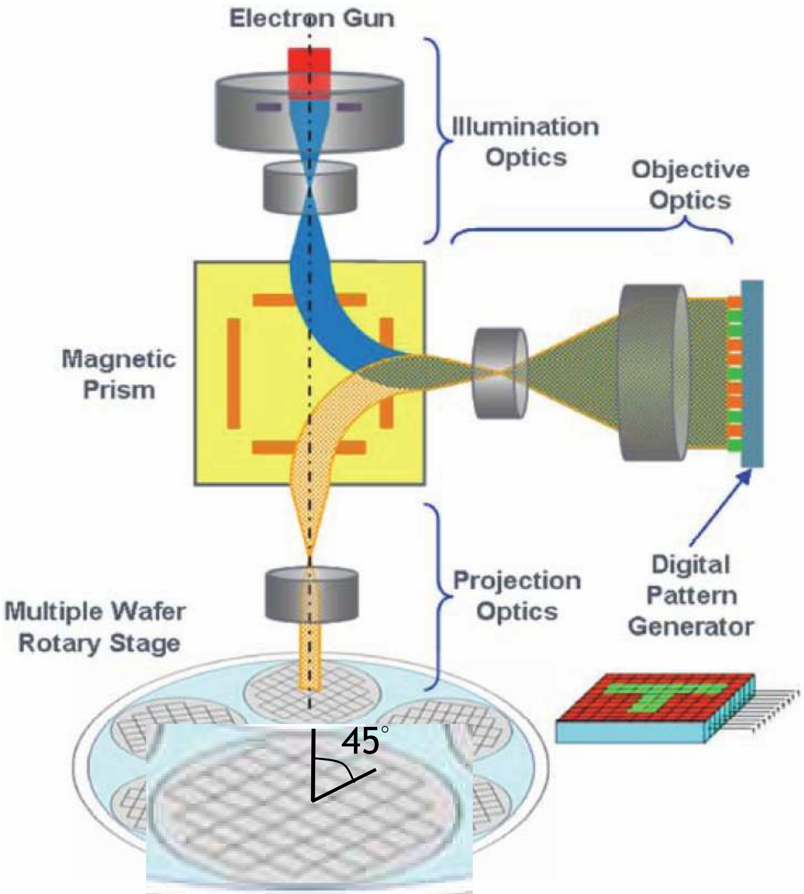
If (a = c) then ? = b else ? = c

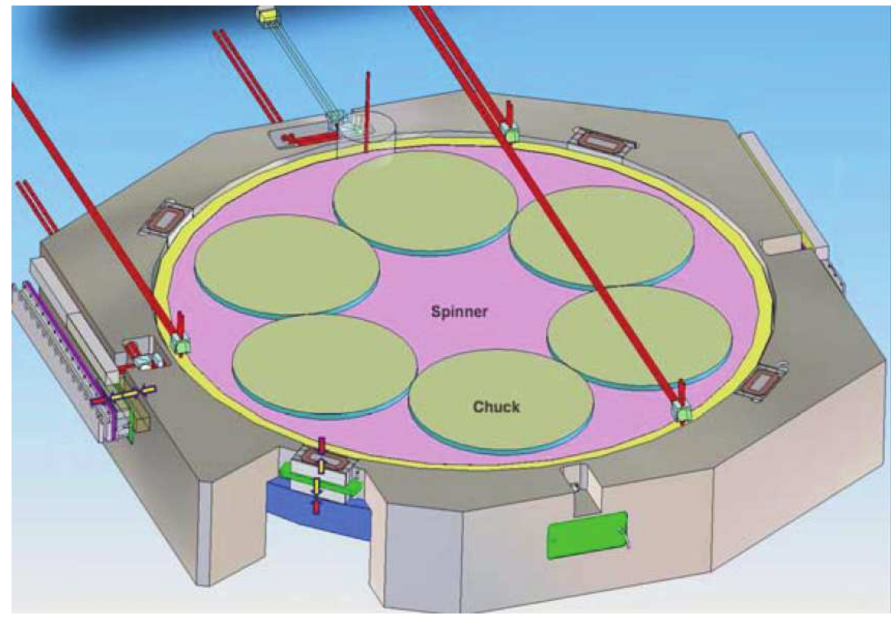
- 1-D region growing can be solve in polynomial time
  - A better solution for complex segmentation maps

Improve Compression Efficiency

- For linear writing system and ASIC layout images → average CR > 10
- For different writing system or compact layout → modify encoding scheme to improve compression efficiency
  - REBL system

REBL Direct-Write Lithography System

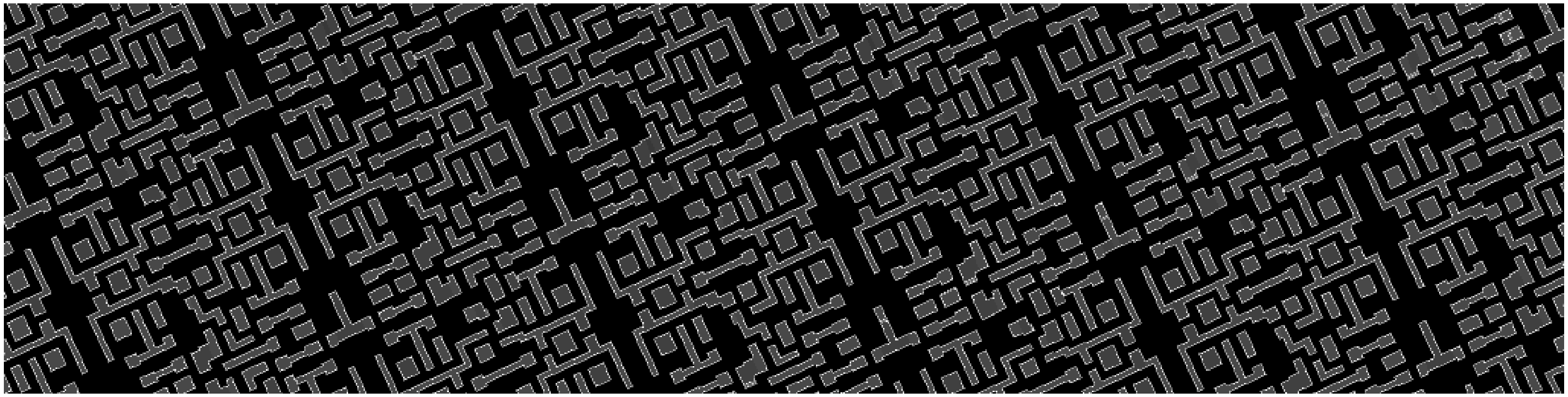




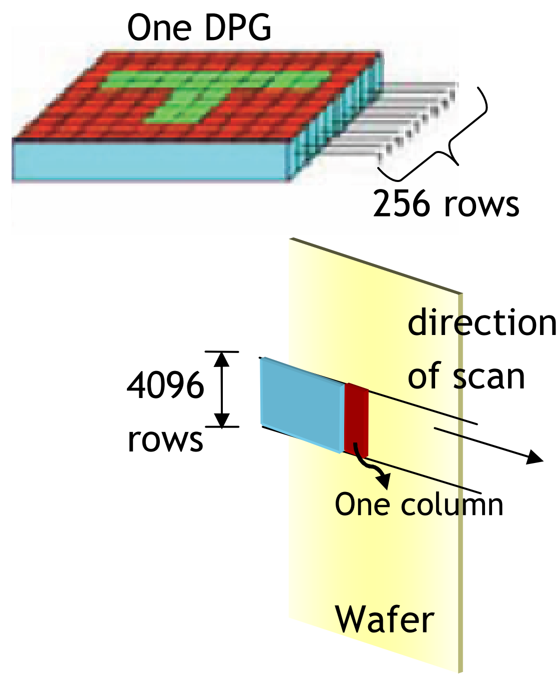
[P. Petric et. al., KLA-Tencor, 08]

- Rotary writer → spiral writing
- 45° between the radius of the stage and the die

REBL Layout Image

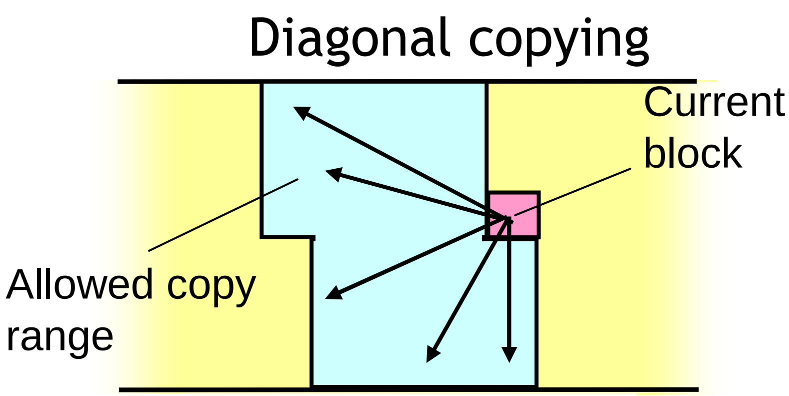


- Layout pattern created by digital pattern generator (DPG)
  - 256 rows per DPG, 16 DPGs in total
  - Column by column writing mechanism
- Layout angle orientation: 15° to 75°
  ➔ ±30° + 45°
- E-beam proximity corrected



Lossless Compression Algorithm for REBL- Block RGC3

- Allow diagonal copying
- Reduce block size and dimension
- Apply 1-D region growing to reduce numbers of regions
- Increase memory size
- Encoding complexity



$$O\left[\frac{mn}{HW}(d_x \times d_y)\right]$$

Memory size= d_x x d_y

Compression Results

	Block GC3			Block RGC3			ZIP	BZip2	JPEG-LS
Buffer size	1.6KB	20KB	40KB	1.6KB	20KB	40KB	32KB	900KB	2.2KB
Block size	4x4	4x4	4x4	5x3	5x3	5x3			
Layout size									
2048x64	3.13	3.37	3.44	4.92	6.54	6.60	3.23	3.95	0.95
1024x256	3.19	3.30	3.36	5.09	6.91	7.12	3.37	4.48	0.96
2048x256	3.19	3.30	3.37	5.10	7.01	7.29	3.43	4.68	0.97

25° Metal 1 layout

- Block RGC3 outperforms Block GC3 and others
- Larger buffer size, larger image size → better compression ratio
- 50 – 69% of improvement due to diagonal copying
  - more effective as buffer size increases

	Block RGC3, 4x4 block, 40 KB Buffer	
Image size	H / V Copying	Diagonal Copying
64×2048	3.44	5.22
256×2048	3.37	5.71

Results for Various Wafer Layers

Image size	Buffer size	Metal 1 Memory			Metal 1 Logic	Poly	Via	
		25°	35°	38°	25°	35°	25°	35°
64×2048	1.7KB	4.92	5.37	5.14	--	8.49	13.14	12.67
256×1024	1.7KB	5.09	5.43	5.33	8.55	8.47	13.58	13.17
256×2048	1.7KB	5.10	5.45	5.35	--	8.51	13.62	13.22
64×2048	20KB	6.54	6.68	6.63	--	11.17	15.31	15.40
256×1024	20KB	6.91	7.08	7.11	14.06	12.50	16.14	16.00
256×2048	20KB	7.01	7.20	7.22	--	12.77	16.35	16.22
64×2048	40KB	6.60	6.79	6.71	--	11.91	15.86	16.11
256×1024	40KB	7.12	7.23	7.34	14.87	12.80	17.05	17.27
256×2048	40KB	7.29	7.41	7.50	--	13.17	17.45	17.79

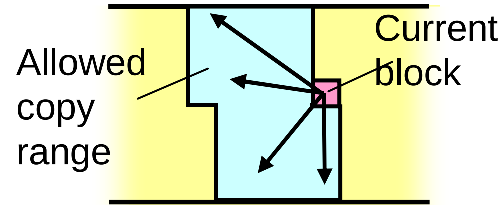
- Higher compression ratio for via than metal 1
- Larger buffer size, larger image size ➔ better compression ratio

Encoding Time

(1) <u>Diagonal copying</u>

- Must compare each image block with each copy distance

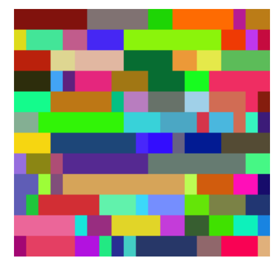
$$O\left(buffer\_size \left(\frac{1}{(block\_size)} + \frac{1}{\beta} \right) \right) \quad , \beta \approx 10$$



(2) <u>Growing regions</u>

- Proportional to avg. # optimal copy distances per block

$$O\left(\frac{\overline{d_{matches,block}}}{block\_size} \right)$$



(3) <u>Combining regions</u>

- Proportional to avg. # optimal copy distances per region
- Inversely proportional to # of blocks per region $\overline{N}$

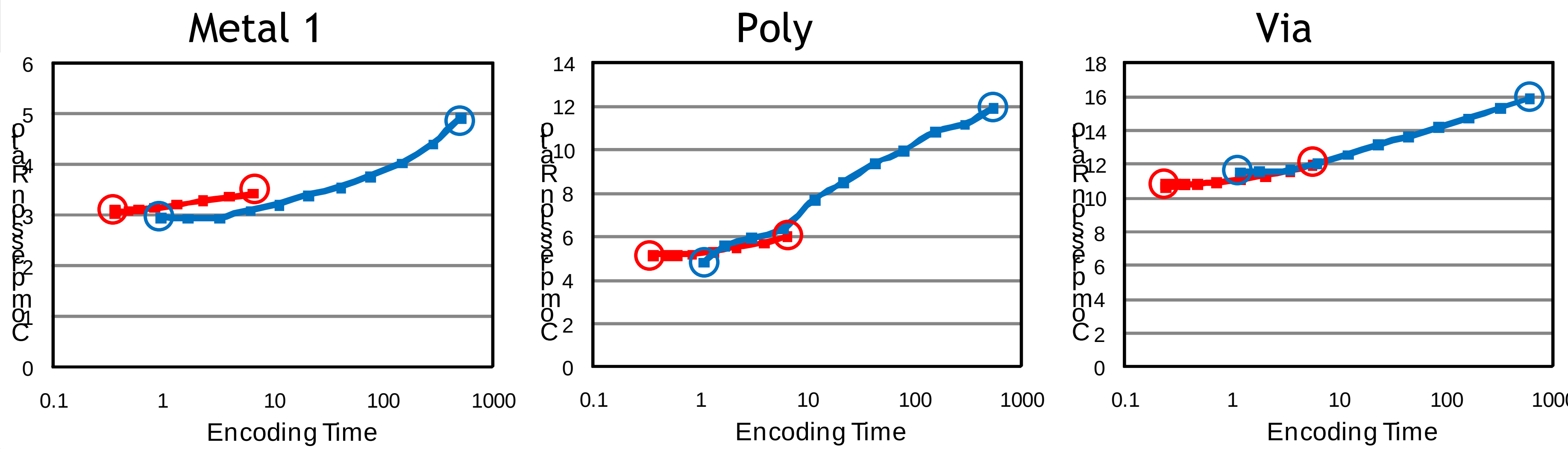
$$O\left(\frac{\overline{d_{matches,region}}}{\overline{N} \cdot block\_size} \right) = O\left(\frac{\overline{d_{matches,region}}}{\overline{region\_size}} \right)$$

Encoding Times

		Diagonal copying		Region-growing		Combining regions		Total encoding time (seconds)	
Image size	Buffer size	Metal1 25°	Via 25°	Metal1 25°	Via 25°	Metal1 25°	Via 25°	Metal1 25°	Via 25°
64×2048	20KB	95.4%	85.5%	4.3%	13.0%	0.5%	1.4%	37.0	41.4
256×1024	20KB	95.2%	85.1%	4.2%	13.8%	0.4%	1.1%	92.1	109.2
64×2048	40KB	96.1%	84.9%	3.6%	14.0%	0.03%	1.1%	66.2	78.7
256×1024	40KB	95.6%	81.1%	4.0%	18.0%	0.02%	0.9%	173.9	226.9

- Dominant factor → Diagonal copying for best copy distance
- Encoding time proportional to buffer size, image size

Encoding Time vs. Compression Ratio



Encoding time normalized to microsecond per pixel

- Smaller buffer size ➔ lower CR and lower encode time
- Block RGC3 additional encode complexity justifiable if higher compression ratios are needed:
  - Metal 1: Higher than 3.5
  - Poly: Higher than 6
  - Via: Higher than 12

Integrating Block GC3 with Writer Systems

- Need to modify the algorithm to achieve best compression efficiency
  - May increase encoding complexity
  - Remain same decoding structure
  - Remain asymmetric compression algorithm

Summary

- Block GC3 solves data delivery problem for direct-write lithography systems
- Implement Block GC3
  - Block GC3 reduces: I/O data rate, System power } → the goal
- Block RGC3 improves compression ratio for REBL system
  - Increase encoder complexity
  - Decoder complexity remains low