Edge Detector Hardware Accelerator

Group 9

Mariana Santos — s233360
Zhuoran Cui — s232207

Naïve Approach

- Fetching data from memory and saving only on 4-byte registers.
- **2 reads per 1 write ➜** each pixel is read **twice** (except in borders)

Naïve Approach

Read n. 1

Register A →

Register B →

Register C →

92	108	82	92	85	35	45	86	99
116	116	98	136	145	93	65	106	67
74	57	52	120	157	124	66	113	65
132	86	44	68	94	93	46	100	54

0	0	0	0	0	0
—	s2	s3	??	?	?
?	?	?	?	?	?

← Pixels read from main memory

92 108 ← Pixels saved on registers

Naïve Approach

Read n. 2

Register A →
Register B →
Register C →

92	108	82	92	85	35	45	86	99
116	116	98	136	145	93	65	106	67
74	57	52	120	157	124	66	113	65
132	86	44	68	94	93	46	100	54

0	0	0	0	0	0
0	s2	s3	s4	?	?
?	?	?	?	?	?

← Pixels read from main memory

92 108 ← Pixels saved on registers

Naïve Approach

Read n. 2

Register A →	92	108	82	92	85	35	45	86	99
Register B →	116	116	98	136	145	93	65	106	67
Register C →	74	57	52	120	157	124	66	113	65
	132	86	44	68	94	93	46	100	54

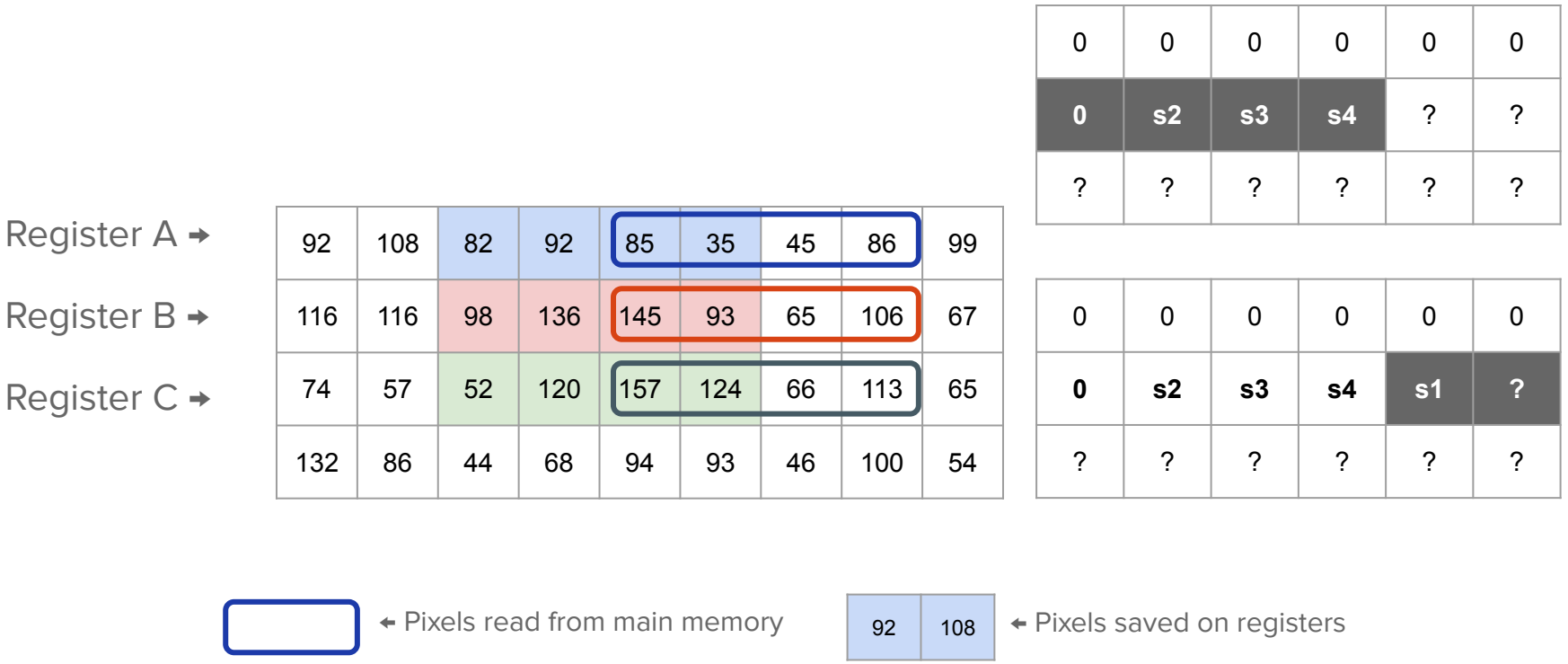
0	0	0	0	0	0
0	**s2**	**s3**	**s4**	?	?
?	?	?	?	?	?

0	0	0	0	0	0
0	**s2**	**s3**	**s4**	**s1**	**?**
?	?	?	?	?	?

← Pixels read from main memory

92 108 ← Pixels saved on registers

Naïve Approach



Naïve Approach

0	0	0	0	0	0
0	**s2**	**s3**	**s4**	?	?
?	?	?	?	?	?

0	0	0	0	0	0

← Pixels read from main memory

92	108

← Pixels saved on registers

Naïve Approach

Read n. 3

(start all over again)

Register A →	92	108	82	92	85	35	45	86	99	0	0	0	0	0	0
Register B →	116	116	98	136	145	93	65	106	67	**0**	**s2**	**s3**	**s4**	**s1**	**s2**
Register C →	74	57	52	120	157	124	66	113	65	?	?	?	?	?	?
	132	86	44	68	94	93	46	100	54						

← Pixels read from main memory

92 108 ← Pixels saved on registers

Current State — Performance

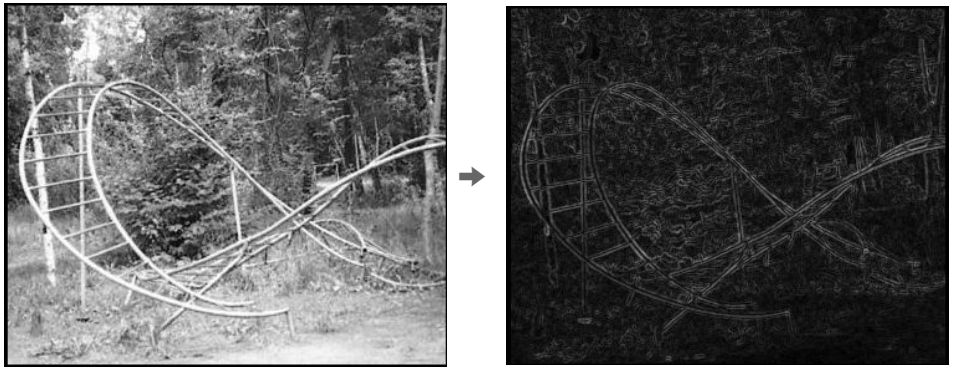
In average, a new 4-byte word will be written every **11 states**.

Considering:
- the clock period to be 80,000 ps
- the edge pixels to not be computed

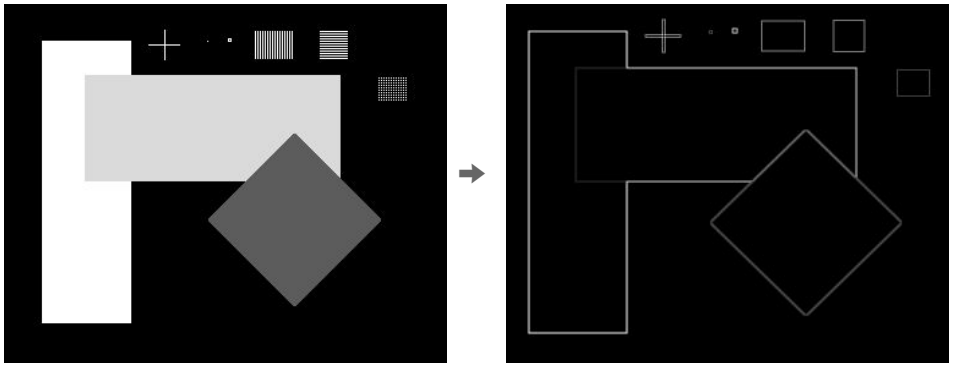
⇒ It should take about 11 * 80.000 * (351 * 287) / 4 ps = 22.162.140.000 ps ≃ **22,162 ms** for an image to be processed.

⇒ It should be able to process 1000/22,162 ≃ **45 images per second**.

Current State — Performance



Our estimation:
22.162.140.000 ps ≃ **22,162 ms**



Simulation takes:
22.056.640.000 ps = **22,057 ms**