An Overlay Architecture for Pattern Matching

Rasha Karakchi, Charles Danial, Jason Bakos

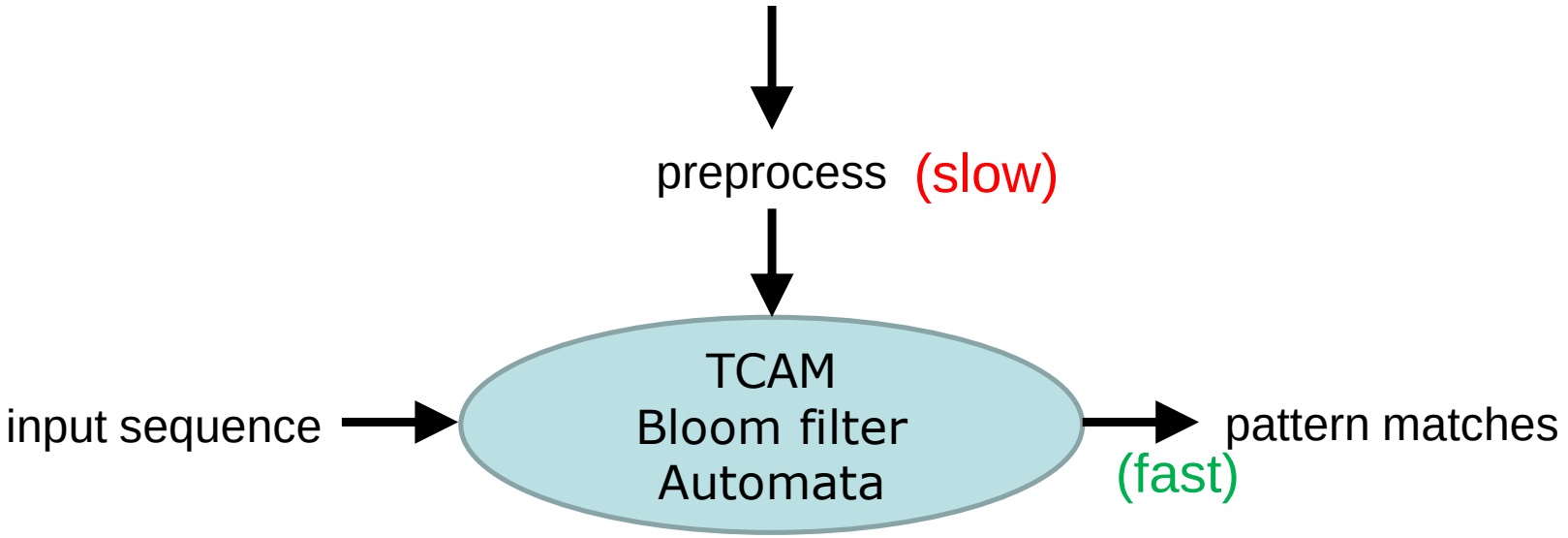
Computer Science and Engineering

UNIVERSITY OF SOUTH CAROLINA

Grant No. 1421059

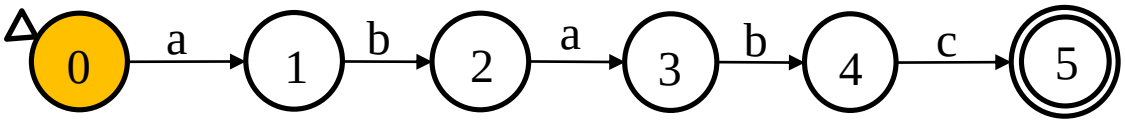
Pattern Matching

Patterns: ex. threat signatures,
network addresses,
genomic seq.



Nondeterministic Finite Automata (NFA)

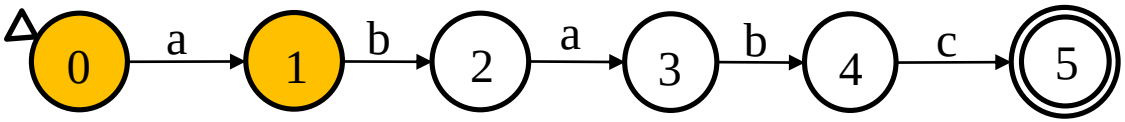
- Recognize pattern: "ababc"



Input	Active States
	0

Nondeterministic Finite Automata (NFA)

- Recognize pattern: "ababc"
- Input: "a"

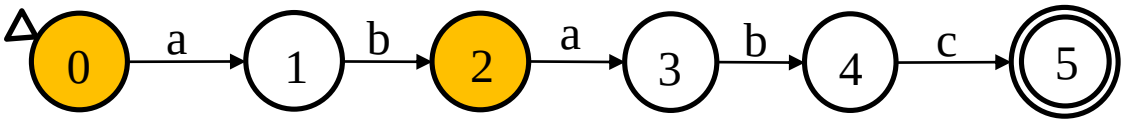


tracking pattern "a"

Input	Active States
	0
a	0,1

Nondeterministic Finite Automata (NFA)

- Recognize pattern: "ababc"
- Input: "ab"

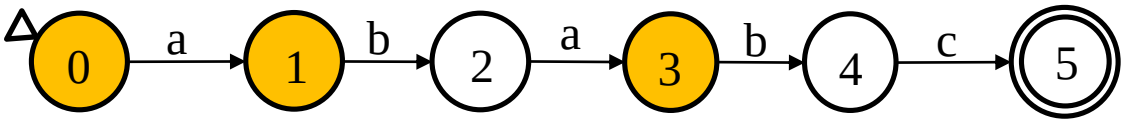


tracking pattern "ab"

Input	Active States
	0
a	0,1
b	0,2

Nondeterministic Finite Automata (NFA)

- Recognize pattern: "ababc"
- Input: "aba"



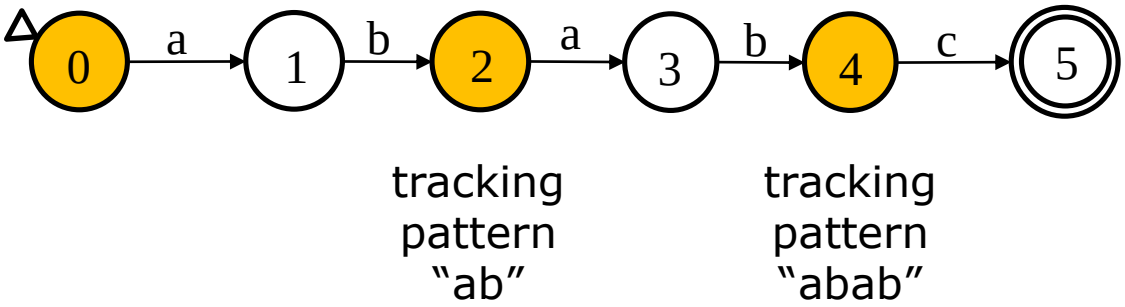
tracking pattern "a"

tracking pattern "aba"

Input	Active States
	0
a	0,1
b	0,2
a	0,1,3

Nondeterministic Finite Automata (NFA)

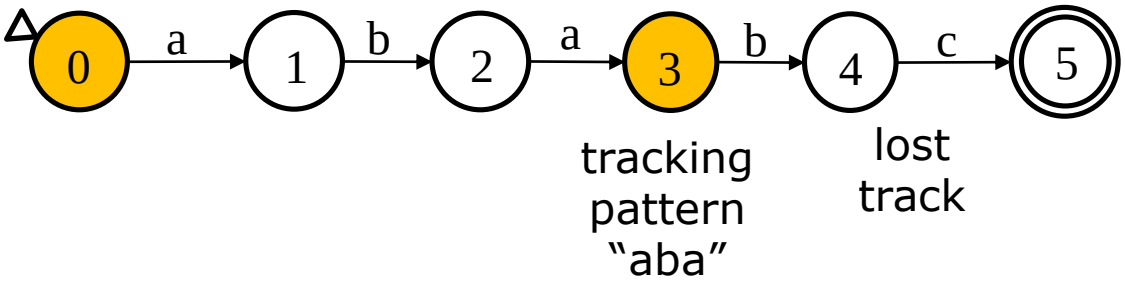
- Recognize pattern: "ababc"
- Input: "abab"



Input	Active States
	0
a	0,1
b	0,2
a	0,1,3
b	0,2,4

Nondeterministic Finite Automata (NFA)

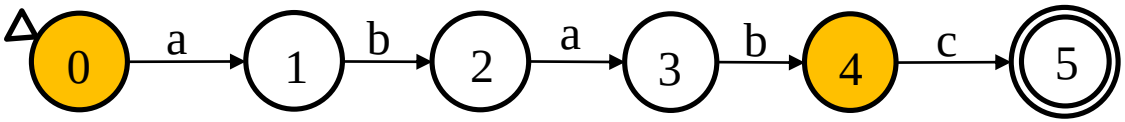
- Recognize pattern: "ababc"
- Input: "ababa"



Input	Active States
	0
a	0,1
b	0,2
a	0,1,3
b	0,2,4
a	0,3

Nondeterministic Finite Automata (NFA)

- Recognize pattern: "ababc"
- Input: "ababab"

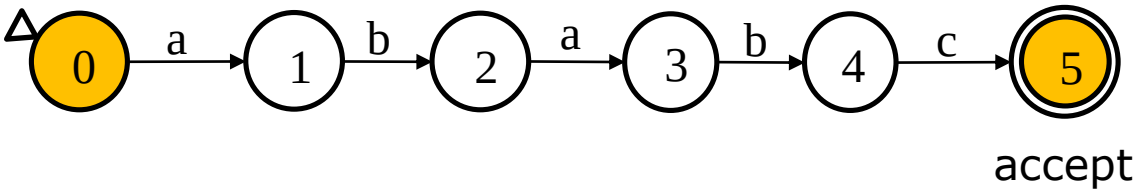


tracking pattern "abab"

Input	Active States
	0
a	0,1
b	0,2
a	0,1,3
b	0,2,4
a	0,3
b	0,4

Nondeterministic Finite Automata (NFA)

- Recognize pattern: "ababc"
- Input: "abababc"

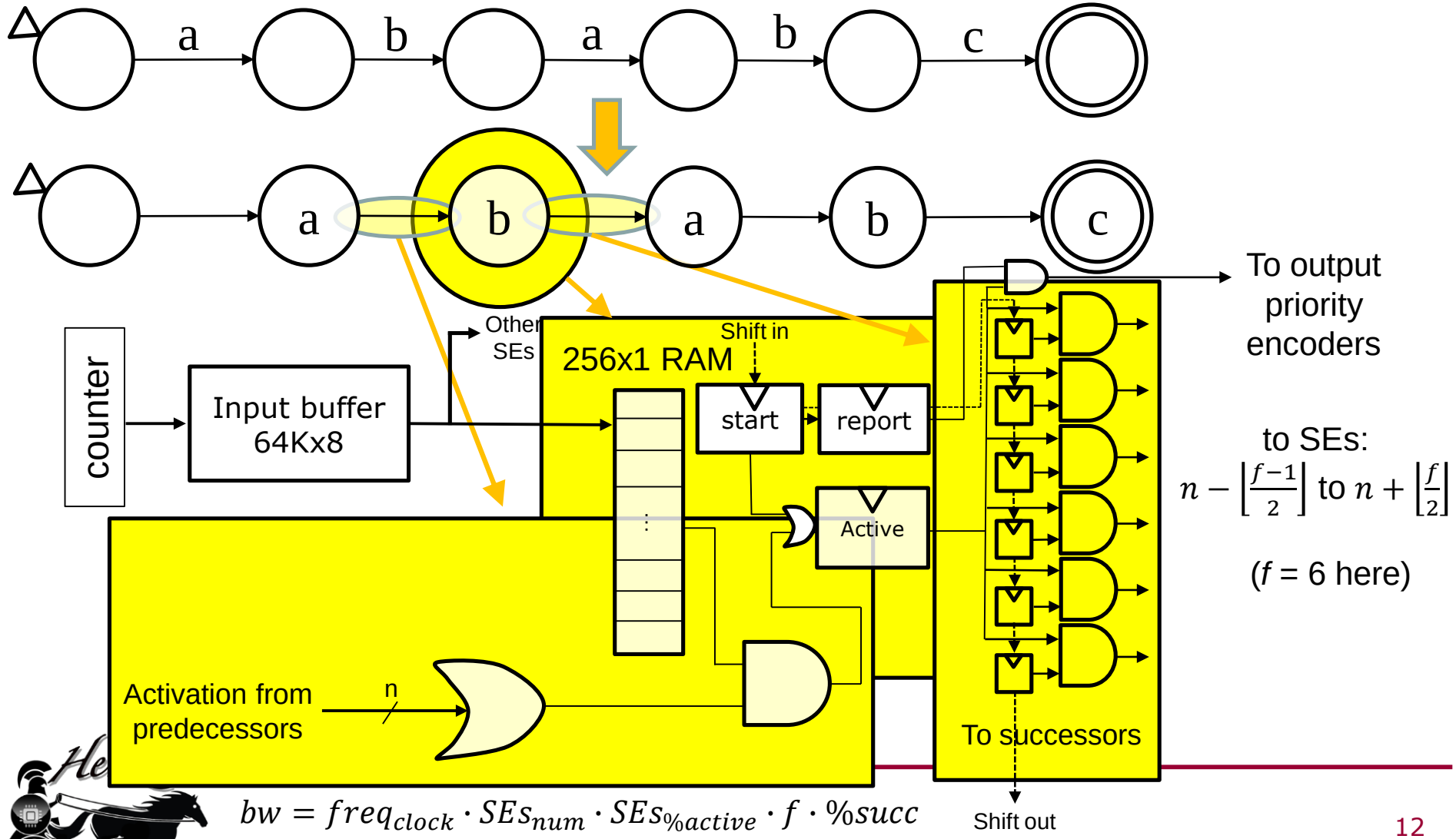


Input	Active States
	0
a	0,1
b	0,2
a	0,1,3
b	0,2,4
a	0,3
b	0,4
c	0,5 (accept)

Applications

- Hamming distance
- Brill tagging rules
- Protein Motif signatures
- Sequential Pattern Mining

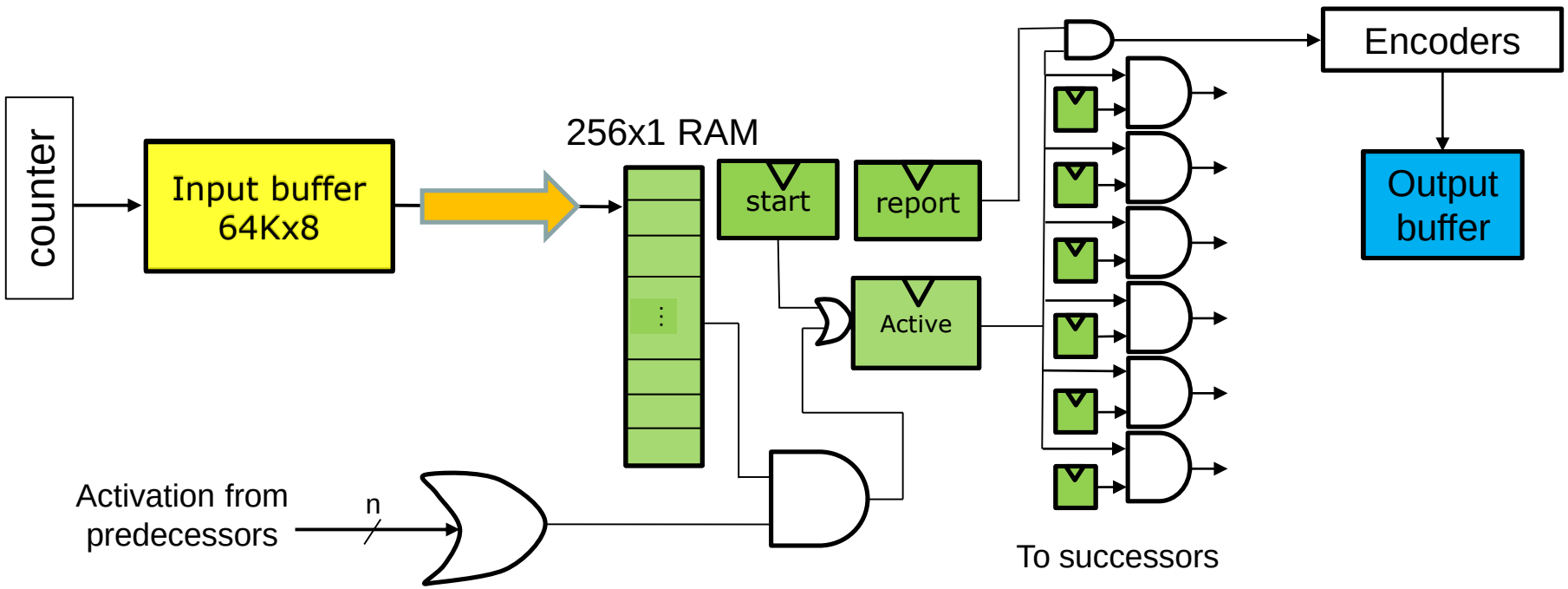
State Element (SE)

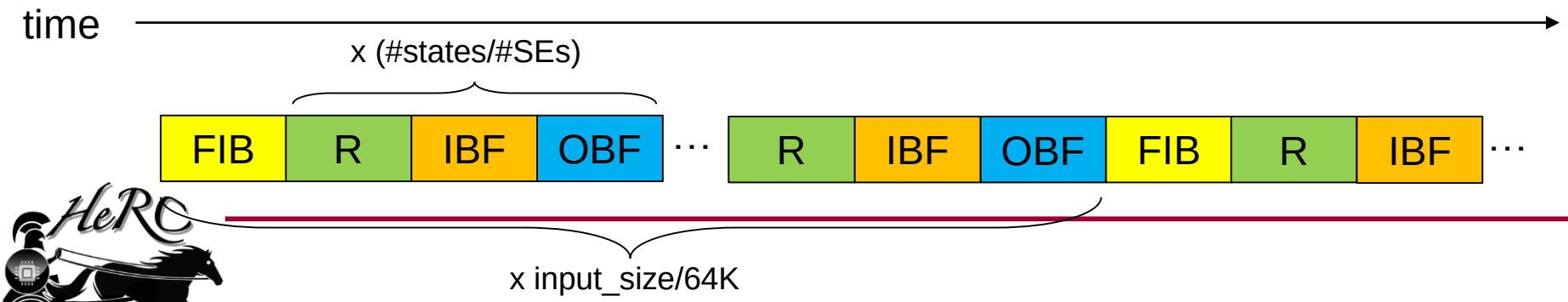


$$bw = freq_{clock} \cdot SEs_{num} \cdot SEs_{\%active} \cdot f \cdot \%succ$$

Runtime Behavior

FIB = fill input buf. | **R** = reconfigure | **IBF** = input buf. flush | **OBF** = output buf. flush





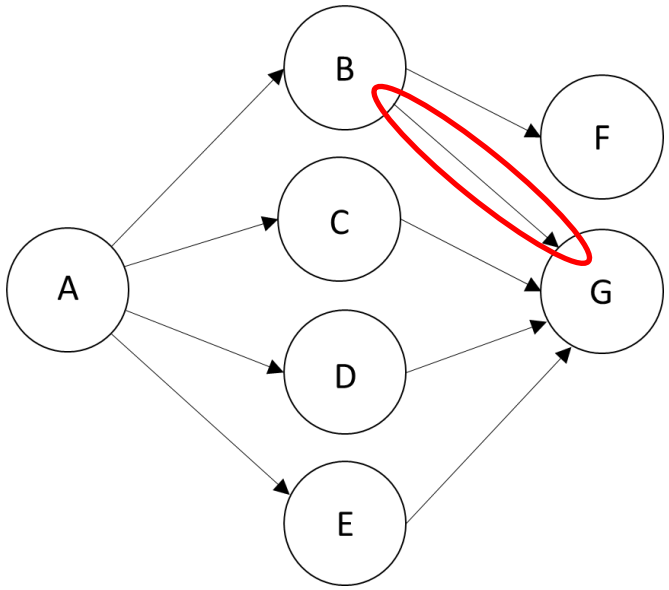
Overlay Configurations

SEs (K)	f	Fmax (MHz)	Max. B/W* for 25% a.s. (GB/s)	Encoders	Max. Report cycles	Max. report rate (GHz)	R (μs)	Through-Put 24K states (MB/s)	Through-Put 128K states (MB/s)
4	103	152	1866	16	100%	2.4	21	14	3
8	44	136	1427	32	50%	2.2	31	27	5
12	25	122	1091	48	33%	2.0	43	32	6
16	12	121	692	64	25%	1.9	53	36	9
20	6	119	426	80	20%	1.9	67	31	9
24	3	112	240	96	17%	1.8	74	67	11

Physical Mapping

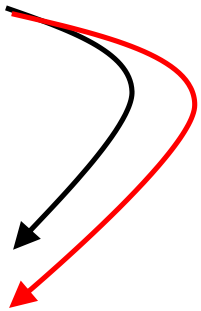
Hardware fan-out *f* allows connection from SE n to SEs:

$$n - \left\lfloor \frac{f-1}{2} \right\rfloor \text{ to } n + \left\lfloor \frac{f}{2} \right\rfloor$$



- Assume f = 9
  - [n-4 to n+4]
- SE Mapping:

State	SE
A	0
B	1
C	2
D	3
E	4
F	5
G	6



- 7! = 5040 possible mappings

f	valid mappings
5	0
6	24 (0.5%)
7	48 (1%)
8	372 (7%)

Performance Results

Benchmark	# states	Minimum Hardware Fan-out Achieved	Overlay	#R/ buffer	Through put (MB/s)
Brill	26668	40	8K	4	**20**
ClamAV	49538	18	12K	5	**13**
Levenshtein	2784	17	12K	1	**63**
Hamming	11346	21	12K	1	**63**
SPM	100500	8	16K	5	**10**
EntityResolution	95136	62	4K	19	**3**
RandomForest	75340	12	16K	5	**15**
PowerEN	40513	29	8K	5	**16**
Snort	69029	60	4K	17	**5**
Fermi	40783	8	16K	2	**24**
Protomata	42061	42	8K	6	**13**
DotStar	96438	4	20K	5	**12**

* iNFAnt on Nvidia Titan Xp ** Nmcart, 4 threads on i5-4440@3.1 GHz

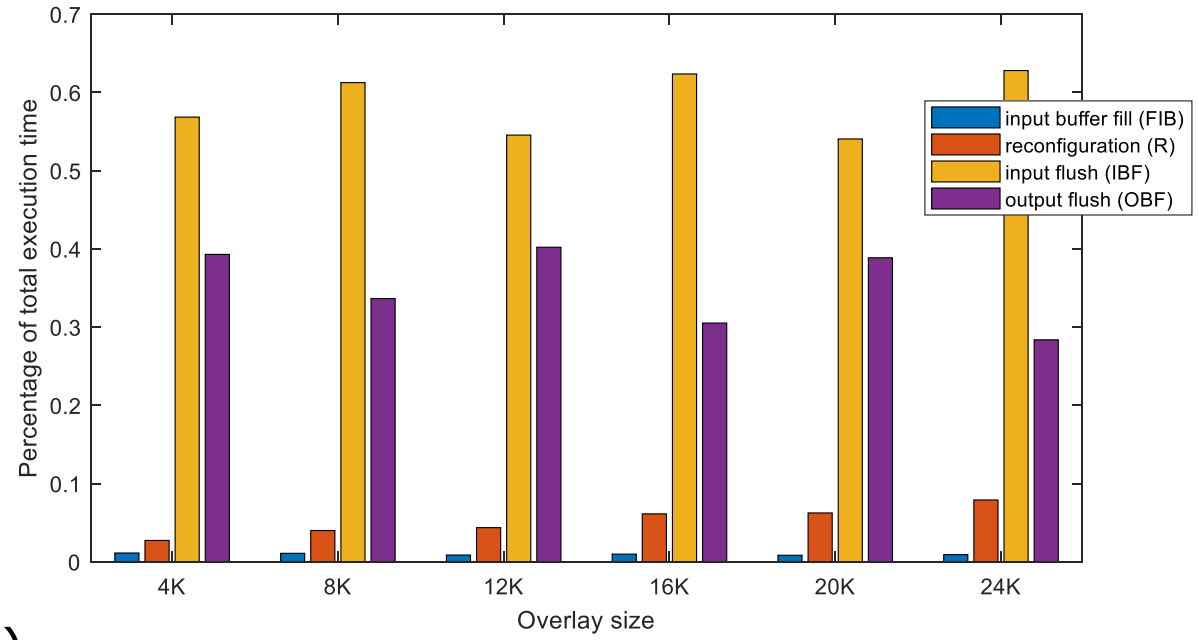
Performance Results

Benchmark	# states	Through put (MB/s)	*Ave. act. states*	iNFAnt (GPU) *	Hyperscan (CPU) **	Speedup
Brill	26668	**20**	*14*	7	1	**3**
ClamAV	49538	**13**	*4*	4	**14**	0.9
Levenshtein	2784	**63**	*88*	38	1	**1.7**
Hamming	11346	**63**	*240*	18	10	**3.5**
SPM	100500	**10**	*6331*	0.5	0.1	**20**
EntityResolution	95136	**3**	*10*	**4**	1	0.8
RandomForest	75340	**15**	*968*	2	0.5	**7.5**
PowerEN	40513	**16**	*31*	**53**	10	0.3
Snort	69029	**5**	*98*	**14**	0.4	0.4
Fermi	40783	**24**	*3854*	*2*	*1*	***12***
Protomata	42061	**13**	*19*	5	1	**2.6**
DotStar	96438	**12**	*3*	**40**	10	0.3

* iNFAnt on Nvidia Titan Xp ** Nmcart, 4 threads on i5-4440@3.1 GHz

Current/Future Work

- Hide input buffer flush latency with output buffer flush
- SAT-solver based mapping algorithm
- Scale up to larger FPGAs and faster memory (DDR4/HBM2)



Questions?

Thank you!