A 135mW Fully Integrated Data Processor for Next-Generation Sequencing

Yi-Chung Wu\textsuperscript{1}, Jui-Hung Hung\textsuperscript{2}, Chia-Hsiang Yang\textsuperscript{1,2}

\textsuperscript{1}National Taiwan University,  
\textsuperscript{2}National Chiao Tung University
Outline

• Next-Generation Sequencing (NGS)
• Data Analyzing Algorithm: sBWT
  – K-Ordered SA Sorting
  – Memory-Efficient Backward Searching
• NGS Data Processor
  – Load-Balanced Splitters Selection
  – Massively-Parallel Sorting Architecture
  – Design Parameters Optimization
• Chip Implementation
• Summary and Conclusions
Next-Generation Sequencing

- Increasing demands on genome analysis

[http://www.popsci.com] [http://www.express.co.uk] [http://www.illumina.com]
[http://www.careersinpublichealth.net] [http://www.foodinsight.org]
Challenges of NGS Data Analyzing

3×10^9 bases

Target: AACGGGT

NGS Data Processor
- High throughput
- Low power

<table>
<thead>
<tr>
<th>Platform</th>
<th>Execution Time</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>&gt;12H</td>
<td>50-150W</td>
</tr>
<tr>
<td>GPU</td>
<td>&gt;1H</td>
<td>~200W</td>
</tr>
</tbody>
</table>

CPU: Central Processing Unit
GPU: Graphics Processing Unit

© 2017 IEEE
International Solid-State Circuits Conference

14.8: A 135mW Fully Integrated Data Processor for Next-Generation Sequencing
NGS Data Analyzing – sBWT Algorithm

Reference DNA: CATG A$ $: termination

K-ordered SA Sorting → FM-Index → Backward Searching → Exact Locations

3×10^9 bases

SA Suffix
0 CATG A$
1 ATGA$C
2 TGA$CA
3 G$CAT
4 A$CATG
5$ CATGA

Splitter Reordering

A$ CATG
GA$ CAT

Suffix Grouping

B0
B1
B2

Sorted Suffixes

$ CATG A$
A$ CAT C
ATGA$C
CATG A$
G$ CAT
GA$ CAT
TGA$CA

[C.-H. Chang, Bioinformatics, 2016]
K-ordered Suffix Array (SA) Sorting

Reference DNA: CATGAS $: termination

3x10^9 bases

K-ordered SA Sorting → FM-Index → Backward Searching → Exact Locations

<table>
<thead>
<tr>
<th>SA</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CATGAS $</td>
</tr>
<tr>
<td>1</td>
<td>ATGAS $C</td>
</tr>
<tr>
<td>2</td>
<td>TGAS $CA</td>
</tr>
<tr>
<td>3</td>
<td>GASCAT</td>
</tr>
<tr>
<td>4</td>
<td>A$CATG</td>
</tr>
<tr>
<td>5</td>
<td>$CATGA</td>
</tr>
</tbody>
</table>

Splitter Reordering

A$CATG
GASCAT

Sorted Suffixes

$CATGAS
A$CATG
ATGAS $C
CATGAS $
GASCAT
TGAS $CA

Suffix Grouping

B0
B1
B2

Suffix Sorting

[min] $CATGAS
A$CATG
ATGAS $C
CATGAS $
GASCAT
TGA$CA

[max]

[C.-H. Chang, Bioinformatics, 2016]
K-ordered Suffix Array (SA) Sorting

Reference DNA: CATGAS $: termination

3x10^9 bases

K-ordered SA Sorting → FM-Index → Backward Searching → Exact Locations

Sorter Reordering

_sorted suffixes_

[SA]

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Sorted Suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CATGAS</td>
<td>$CATGAS</td>
</tr>
<tr>
<td>1 ATGASC</td>
<td>ATGASC</td>
</tr>
<tr>
<td>2 TGA$CA</td>
<td>TGA$CA</td>
</tr>
<tr>
<td>3 GA$CAT</td>
<td>GA$CAT</td>
</tr>
<tr>
<td>4 A$CATG</td>
<td>A$CATG</td>
</tr>
<tr>
<td>5 $CATGA</td>
<td>$CATGA</td>
</tr>
</tbody>
</table>

[C.-H. Chang, Bioinformatics, 2016]
Ferragina and Manzini (FM) Index

Reference DNA: CATG A $ $: termination

K-ordered SA Sorting → FM-Index → Backward Searching → Exact Locations

- Memory efficient
- Invertible

3x10^9 bases

[CM. Chang, Bioinformatics, 2016]
Ferragina and Manzini (FM) Index

- Memory efficient
- Invertible

Reference DNA: \texttt{CATGAS} \texttt{$$}: termination

K-ordered SA Sorting → FM-Index → Backward Searching → Exact Locations

<table>
<thead>
<tr>
<th>C</th>
<th>SA</th>
<th>F</th>
<th>L</th>
<th>OCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>$</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>A</td>
<td>$</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>A</td>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>C</td>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>GA</td>
<td>$</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>TGA</td>
<td>A</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Sorted Suffixes
- \texttt{$$CATGAS}$
- \texttt{A$CATGA$}
- \texttt{ATGAS$C$}
- \texttt{CATGAS}$
- \texttt{GAS$CAT$}
- \texttt{TGAS$CA$}

© 2017 IEEE International Solid-State Circuits Conference

14.8: A 135mW Fully Integrated Data Processor for Next-Generation Sequencing

Backward Searching Scheme

Reference DNA: $\text{CATGAS} \quad \$: termination

Target String: GA
Searched from the last letter

Initialization:
$\text{Index}_{\text{min}} = 1$
$\text{Index}_{\text{max}} = 5$

LF Mapping

$\text{Index}_{\text{min}} = C[S[i]] + \text{OCC}[\text{Index}_{\text{min}}-1, S[i]] + 1$

$\text{Index}_{\text{max}} = C[S[i]] + \text{OCC}[\text{Index}_{\text{max}}, S[i]]$

Backward Searching Scheme

Reference DNA: $\text{CATGAS}$ $\$: termination

K-ordered SA Sorting $\rightarrow$ FM-Index $\rightarrow$ Backward Searching $\rightarrow$ Exact Locations

3x10^9 bases

Target String: GA

$S = \text{AG}$

$S[1] = \text{A}$

1st Iteration:

$\text{Index}_{\text{min}} = 0+0+1 = 1$

$\text{Index}_{\text{max}} = 0+2 = 2$

**LF Mapping**

$\text{Index}_{\text{min}} = C[S[i]] + \text{OCC}[\text{Index}_{\text{min}}-1,S[i]] + 1$

$\text{Index}_{\text{max}} = C[S[i]] + \text{OCC}[\text{Index}_{\text{max}},S[i]]$

Backward Searching Scheme

Reference DNA: CATGAS $: termination

Target String: GA
S = AG

2nd Iteration:

Index_{min} = 3+0+1 = 4
Index_{max} = 3+1 = 4

LF Mapping

\[
\begin{align*}
\text{Index}_{\text{min}} &= C[S[i]] + \text{OCC}[\text{Index}_{\text{min}} - 1, S[i]] + 1 \\
\text{Index}_{\text{max}} &= C[S[i]] + \text{OCC}[\text{Index}_{\text{max}}, S[i]]
\end{align*}
\]

[C.-H. Chang, Bioinformatics, 2016]
Backward Searching Scheme

3x10^9 bases

Reference DNA: CATGAG $ $: termination

0 1 2 3

K-ordered SA Sorting

FM-Index

Backward Searching

Exact Locations

Target String: GA
S = AG

2^{nd} Iteration:

Index_{min} = 3+0+1 = 4
Index_{max} = 3+1 = 4

LF Mapping

\[
\begin{align*}
\text{Index}_{\text{min}} &= C[S[i]] + \text{OCC}[\text{Index}_{\text{min}} - 1, S[i]] + 1 \\
\text{Index}_{\text{max}} &= C[S[i]] + \text{OCC}[\text{Index}_{\text{max}}, S[i]]
\end{align*}
\]

Overall Architecture

Suffix Array (SA) Sorting

- Splitter Reordering
- Serial Sorter
- Suffix Sorting
- Suffix Grouping

Interconnect Fabric

Backward Searching

- LF Mapping
- Suffix Array Query

Overflow Cache

Splitter Cache
Splitter Reordering

- Up/Down sampling scheme
  - Suffixes are grouped more evenly
  - Reduce sorting complexity
Suffix Sorting

- All suffixes in each bucket must be sorted

![Diagram showing suffix sorting process]

Sorted Suffixes

$ CAT
A $ CA
ATGA
CATG
GASC
TGA$

Unsorted Suffixes

CATG
ATGA
TGAS$
GASC
ASCA
$CAT

Suffix Storing

Suffix Fetching

Serial Sorter

Recursive Sort Control

Overflow Cache

Overflow Detector
Serial Sorter

- High parallelism
- Latency = 1 cycle

[J. Matai, Int. Symp. Field Programmable Gate Arrays, 2016]
Serial Sorting Operation

- Cascaded SEs
- Well ordered after each cycle
Cascaded SEs

Well ordered after each cycle
Backward Searching

- OCC check point = 0
- Number of T in L Column = 1
- Query = 0 + 1 = 1

Target String

Letter Detection

LF Mapping

SA Query

Exact Locations

OCC Table

C | SA | L | OCC
---|----|---|-----
0 | 5  | A | 1000
2 | ×  | G | ××××
3 | ×  | C | ××××
4 | 0  | $ | 1110

Boundary

Index\(_{\text{min}}\)

Index\(_{\text{max}}\)

32

forward

OCC Query

OCC Table

Matching
Optimized-Architecture: Memory

Memory Requirement vs. Sampling Factor

Overall Memory Reduction

Baseline  | Down Sampling

© 2017 IEEE International Solid-State Circuits Conference
Optimized-Architecture: Latency

Search Latency vs. Sampling Factor

Overall Search Latency Reduction

© 2017 IEEE
International Solid-State Circuits Conference
Chip Photograph

- **Technology**: 40nm 1P10M CMOS
- **Chip Size**: 2.80 × 2.80mm²
- **Core Size**: 2.25 × 2.25mm²
- **Gate Count**: 7.4M
- **Core VDD**: 0.9V
- **I/O VDD**: 3.3V
- **Clock Freq.**: 200MHz
- **Power**: 135mW

© 2017 IEEE International Solid-State Circuits Conference
### Performance Comparison

<table>
<thead>
<tr>
<th>Design</th>
<th>AMD Opteron 6128 8-Core</th>
<th>NVIDIA Tesla M2075 448-Core</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>45nm</td>
<td>40nm</td>
<td>40nm</td>
</tr>
<tr>
<td>Chip Area [mm²]</td>
<td>346</td>
<td>520</td>
<td>7.84</td>
</tr>
<tr>
<td>External Memory [GB]</td>
<td>128</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Local Memory [KB]</td>
<td>17,120</td>
<td>N/A</td>
<td>384</td>
</tr>
<tr>
<td>Freq. [MHz]</td>
<td>2,000</td>
<td>1,150</td>
<td>200</td>
</tr>
<tr>
<td>Power [mW]</td>
<td>80,000</td>
<td>200,000</td>
<td>135</td>
</tr>
<tr>
<td>Throughput [suffixes/s]</td>
<td>$6.9 \times 10^4$</td>
<td>$8.3 \times 10^5$</td>
<td>$5.1 \times 10^6$</td>
</tr>
<tr>
<td>Energy Efficiency [suffixes/mJ]</td>
<td>0.87</td>
<td>4.2</td>
<td>$3.7 \times 10^4$</td>
</tr>
<tr>
<td>TAR [suffixes/s/mm²]</td>
<td>$2.0 \times 10^2$</td>
<td>$1.6 \times 10^3$</td>
<td>$6.4 \times 10^5$</td>
</tr>
</tbody>
</table>
Performance Improvement

- Energy Efficiency
  - CPU: 43,065×
  - GPU: 8,971×

- Throughput to Area Ratio (TAR)
  - CPU: 3,208×
  - GPU: 402×

![Graph showing energy efficiency and throughput-to-area ratio for different processors. This Work is marked with a star.]
Summary and Conclusions

• *First* integrated NGS data processor
  – Realizes both SA Sorting and Backward Searching

• Hardware-friendly NGS data analysis algorithm: sBWT
  – K-ordered SA sorting + memory-efficient FM-index

• Optimized architecture
  – Serial Sorting w/ 2,048 SEs $\rightarrow$ 1 cycle latency
  – SA and OCC down-sampling $\rightarrow$ 98.7% memory reduction
  – Backward Searching: 32 matching modules + bidirectional searching $\rightarrow$ 96.7% latency reduction

• Orders of magnitude improvement (vs. high-end GPU)
  – Energy efficiency: >8,900x
  – Throughput-to-area ratio: >400x
References


