Parallel Computation of Simple Arithmetic using Peptide- Antibody Interactions

M. Sakthi Balan
Kamala Krithivasan

Theoretical Computer Science Lab
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai – 600036.

***Email**: sakthi@cs.iitm.ernet.in*
kamala@iitm.ernet.in

Organization

- DNA Computing
- Peptide Computing
- Proposed Model
- Addition Algorithm
- Subtraction Algorithm
- Discussion

DNA Computing

- Uses DNA strands and Watson-Crick Complementarity as operation
- Highly ***non-deterministic***
- Massive ***parallelism***
- Solves NP-Complete Problems quite efficiently

Peptide Computing

- Uses peptides and antibodies
- Operation – binding of antibodies to epitopes in peptides
- ***Epitope*** – The site in peptide recognized by antibody
- Highly ***non-deterministic***
- Massive ***parallelism***

Peptide Computing Contd..

- Peptides – sequence of amino acids
- Twenty amino acids.

  Example – Glycine, Valine
- Connected by covalent bonds

Peptide Computing Contd..

- Antibodies recognizes epitopes by binding to it
- Binding of antibodies to epitopes has associated power called ***affinity***
- Higher priority to the antibody with larger affinity power

Computing
DNA Vs Peptide

- Four building blocks
- Adenine (A), Guanine(G), Cytosine (C), Thiamine (T)
- Only one reverse complement – Watson-Crick Complement
- Complement (A) = T and Complement (G) = C

- Twenty building blocks (20 amino acids)
- Example: Glycine, Valine
- Different antibodies can recognize different epitopes
- Binding affinity of antibodies can be different

Proposed Model

- Consists of a peptide and set of antibodies
- Peptide sequence has n position specific epitopes
- Epitopes $ep_i = y_i x_i z_i$, y_i and z_i are *switching epitopes* for the i^{th} bit.

Peptide Sequence for a 5-bit number

Antibodies

- $\mathcal{A} = \{A_0, A_1, \ldots, A_{n-1}\}$
- $\mathcal{B} = \{B_0, B_1, \ldots, B_{n-1}\}$
- $\mathcal{T}_{\mathcal{AB}} = \{T_{AB0}, T_{AB1}, \ldots, T_{AB(n-1)}\}$
- $\mathcal{T}_{\mathcal{BA}} = \{T_{BA0}, T_{BA1}, \ldots, T_{BA(n-1)}\}$

Binding Sites

For A_i: $A_i \rightarrow y_i x_i$, $A_i \rightarrow x_i$

For B_i: $B_i \rightarrow x_i z_i$, $B_i \rightarrow x_i$

$y_i x_i > x_i$

$x_i z_i > x_i$

$T_{ABi} \longrightarrow Z_i$

$T_{BAi} \longrightarrow y_i$

Affinity

- $aff(T_{ABi}) > aff(A_i)$
- $aff(T_{BAi}) > aff(B_i)$
- $aff(T_{ABi}) = aff(T_{BAi})$

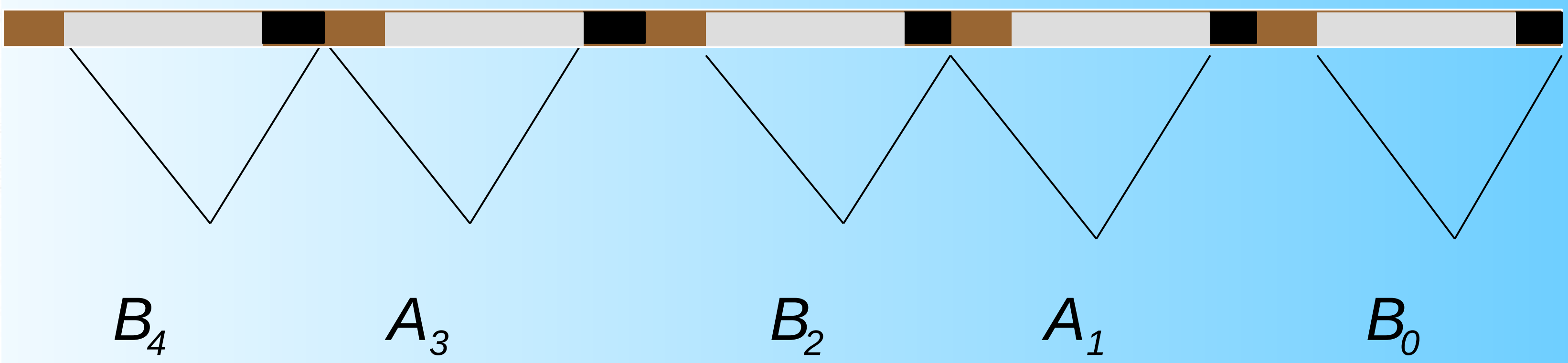
What it denotes?

- A_i – denotes i^{th} bit is zero
- B_i – denotes i^{th} bit is one
- T_{ABi} – used to switch i^{th} bit from zero to one
- T_{BAi} – used to switch i^{th} bit from one to zero

Representation of Binary Numbers

- If the i^{th} bit is 0 then the antibody A_i is bounded to the epitope $y_i x_i$
- If the i^{th} bit is 1 then the antibody B_i is bounded to the epitope $x_i z_i$

Example



10101

Addition of Two Binary Numbers

$A = a_{n-1}a_{n-2}\ldots a_0$ $B = b_{n-1}b_{n-2}\ldots b_0$

$$C = c_n c_{n-1}c_{n-2}\ldots c_0$$

XOR

	a_i	b_i	c_i
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Addition (Contd..)

- First step – guessing equivalent to XOR gate.
- The bit c_n is initialized to zero.
- Carry propagation.

Addition (Contd..)

- Carry occurs only when both the bits a_i and b_i are 1.
- Carry is propagated to the left until both the bits a_j and b_j ($j > i$) are 0.
- If no such j exists then propagation stops making n^{th} bit 1.
- j.j-1….i+1 is called the carry block.
- For each carry block j.j-1….i+1 invert the digits c_k ($i+1 \le k \le j$)

Algorithm

1. Add antibodies A_i where $a_i = 0$ and $b_i = 0$ or $a_i = 1$ and $b_i = 1$.
2. Add antibodies B_i where $a_i = 0$ and $b_i = 1$ or $a_i = 1$ and $b_i = 0$.
3. For all carry block $j_k j_k - 1 \ldots i_k + 1$ do the following in parallel. For $i_k + 1 \leq s \leq j_k$
   a) Add antibodies T_{ABs},
   b) Add antibodies B_s,
   c) Add antibodies T_{BAs}, and
   d) Add antibodies A_s.

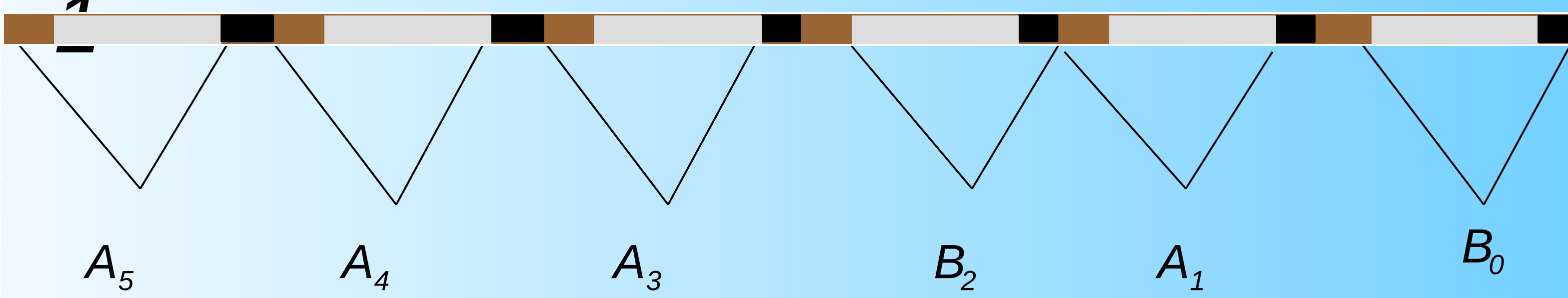
Example

1011
0

+

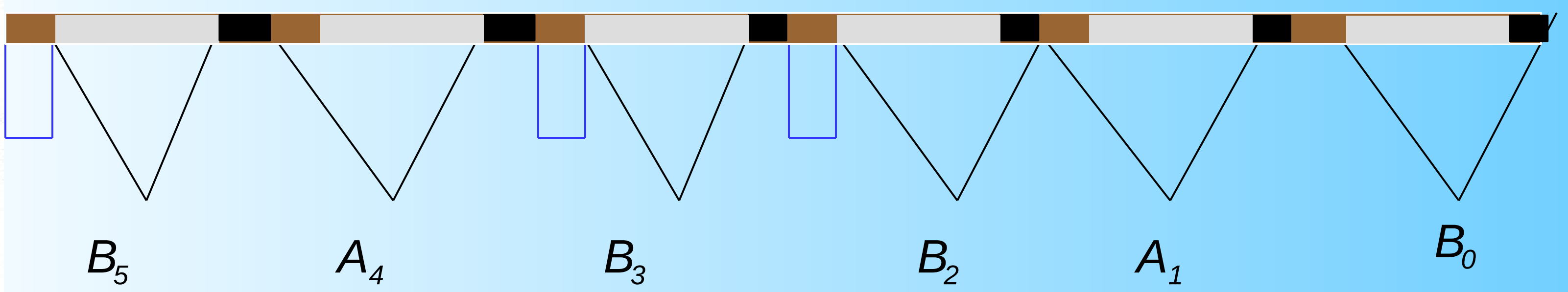
000101

1001
1



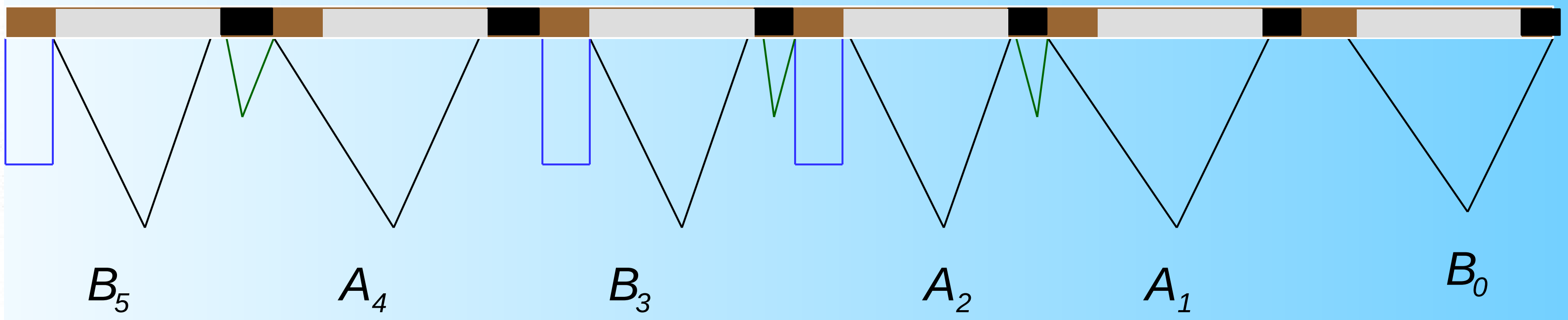
Example (Contd..)

101101



Example (Contd..)

101001



Algorithm

ADD(A,B,C)

1. XOR(A,B,C)
2. BlockInversion(I_1,I_2,…I_k,C) where I_j are carry blocks and k is the number of carry blocks.

Algorithm - Same(C)

To get the peptide sequence with antibodies in workable form

1. Add excess of epitopes y_i
2. Add antibodies A_i
3. Add excess of eptiopes z_i
4. Add antibodies B_i

Algorithm - Subtraction

SUB(A,B,C)

- BlockInversion(I_{1}, B, B') where I_1 = n-1…0
- ADD(B', ONE, B'') where ONE = $a_{n-1}a_{n-2}\ldots a_1 1$, $a_i = 0$
- ADD(A, B'', C)
- Inverttozero(C,n)

Algorithm – Inverttozero

Inverttozero(C,i)

- Same(C)
- Add antibody T_{ABi}
- Add antibody A_i

Discussion

- To extract numbers from this system
  - NMR can be used or
  - X-ray crystallography
- Limitations
  - Obtaining monoclonal antibodies
  - Manual process
- Implementation ?
- Universal operations ?

Acknowledgments

Authors thank

Saravanakumar Narayanan,

TU-Munchen, Germany

for helpful discussions

"They were built by 3 billion years of evolution, and we're just beginning to tap their potential to serve non-biological purposes. Nature has given us an incredible toolbox, and we're starting to explore what we might build"

Leonard Adleman

Thank You