Error Correction/RAID Engine for DNA-Based Storage

STUDENTS:
Firas Ramadan & Muhammad Dahamshi

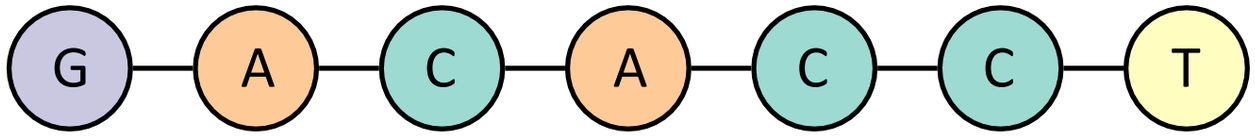
Supervisor : Amit Berman

Motivation & background

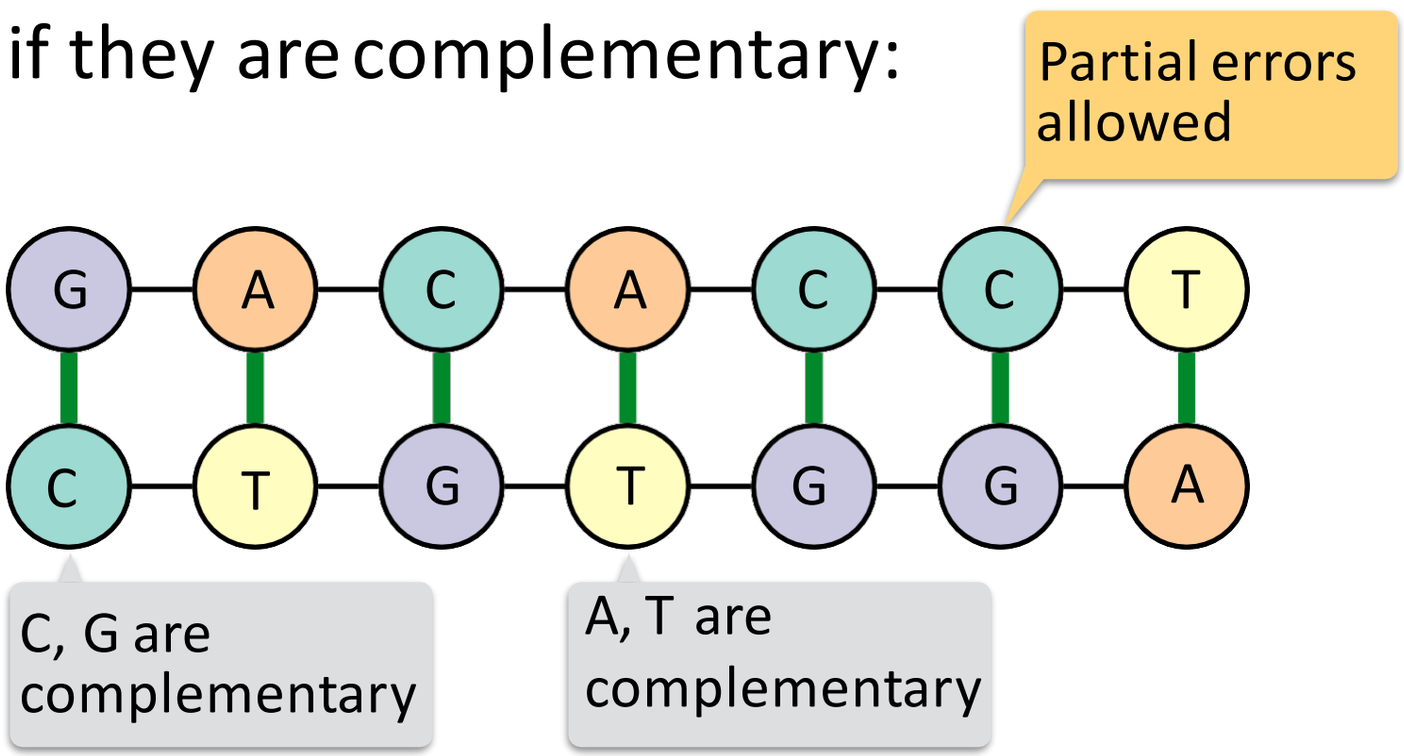
Four nucleotides:

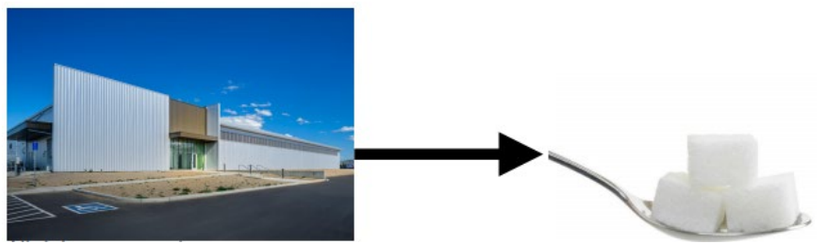
- A — Adenine
- C — Cytosine
- G — Guanine
- T — Thymine

DNA strand (oligonucleotide) is a linear sequence of these nucleotides



Two strands can bind to each other if they are complementary:





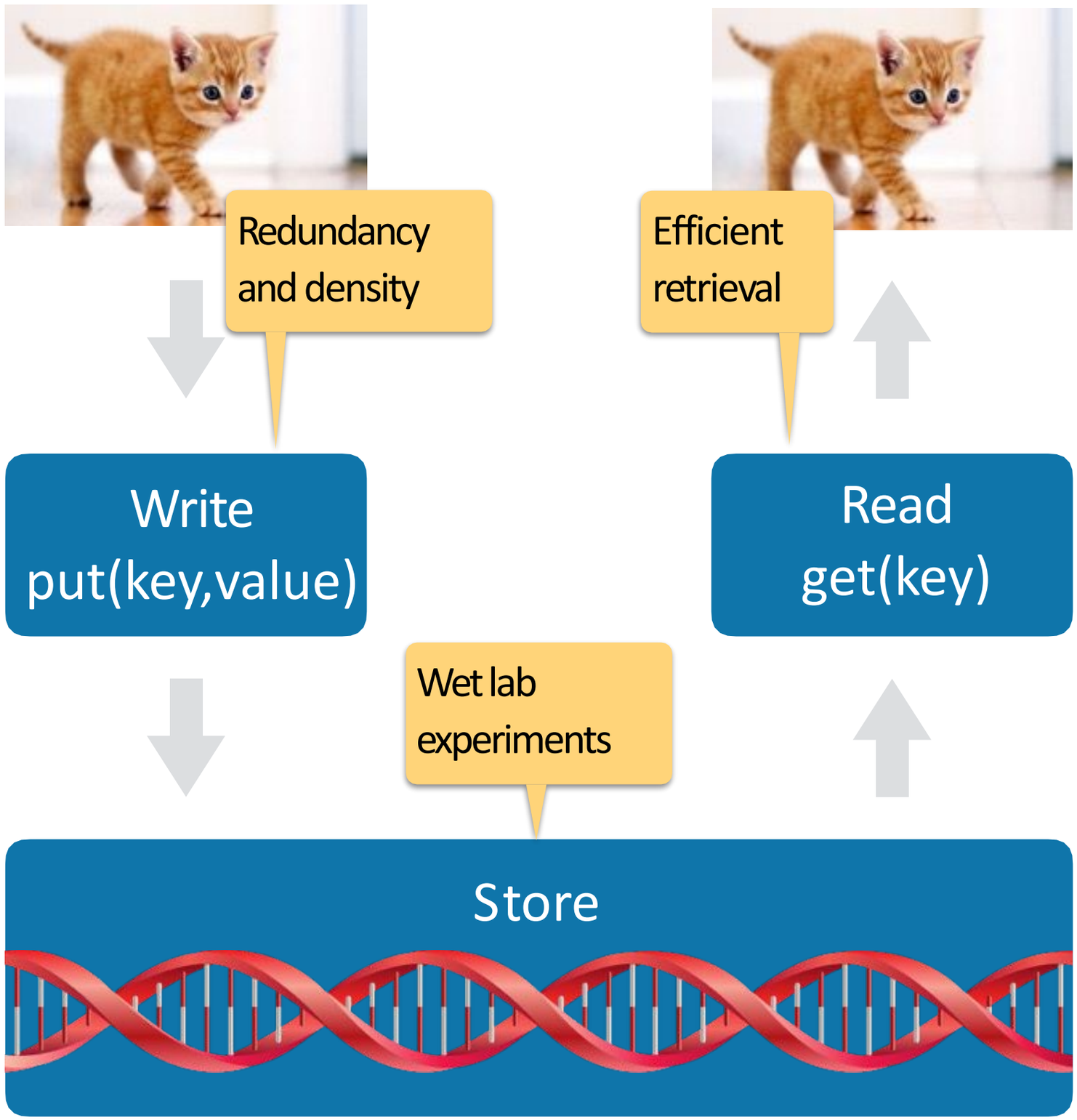
Extremely dense

Theory: 1 exabyte in 1 in^3

Extremely durable

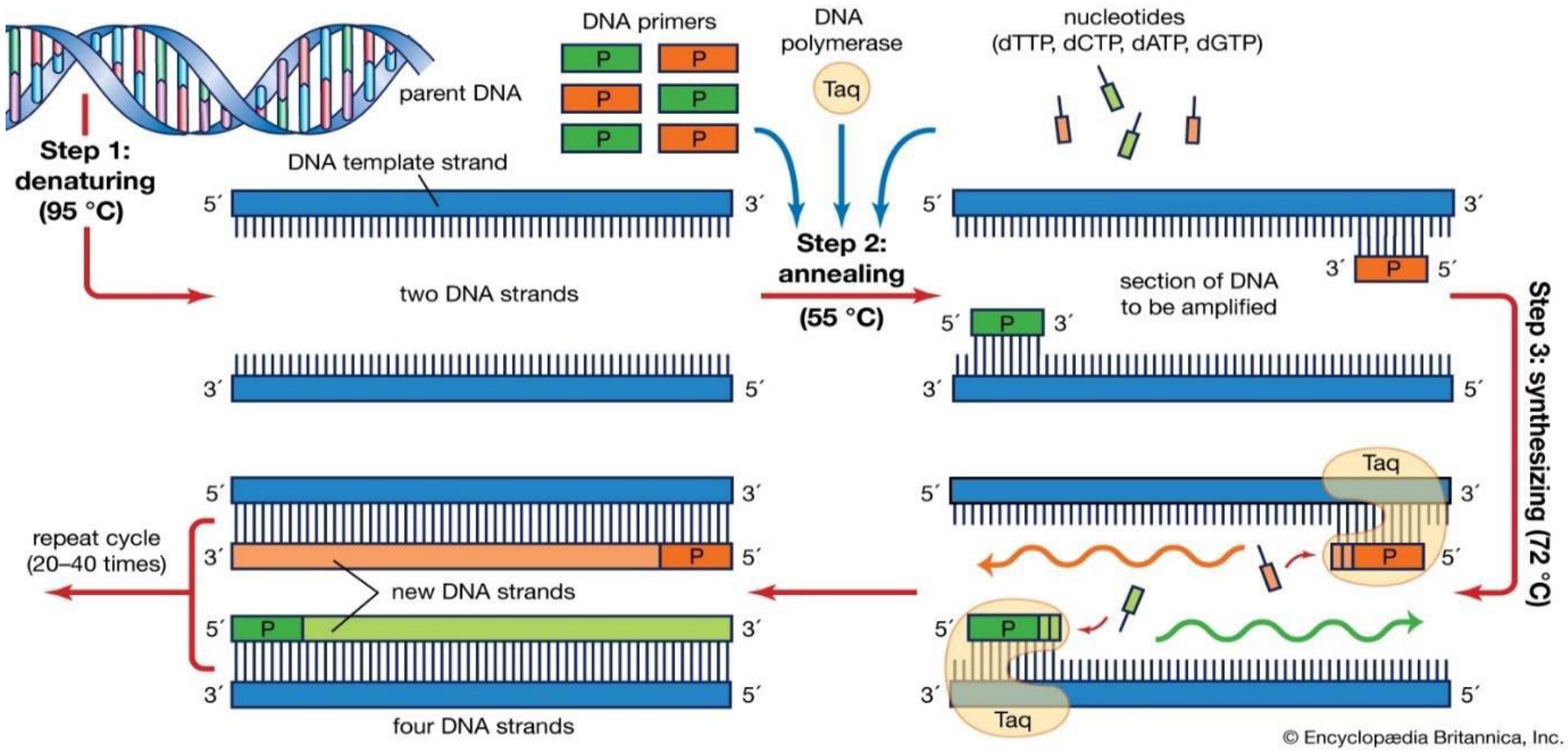
Half life > 500 years

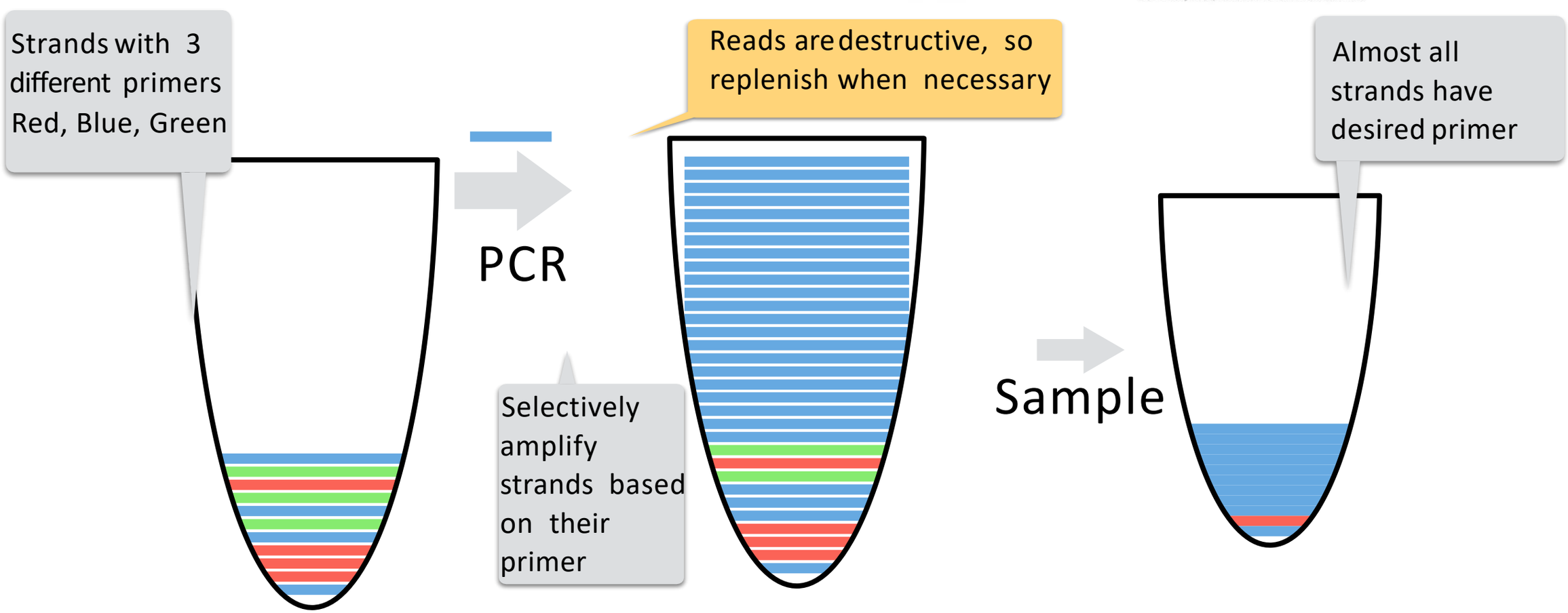
A DNA-based archival storage system



-Reliability-density trade-off
-Increasing Reliability
-Random Access
-Error analysis
-Model of truncated strands

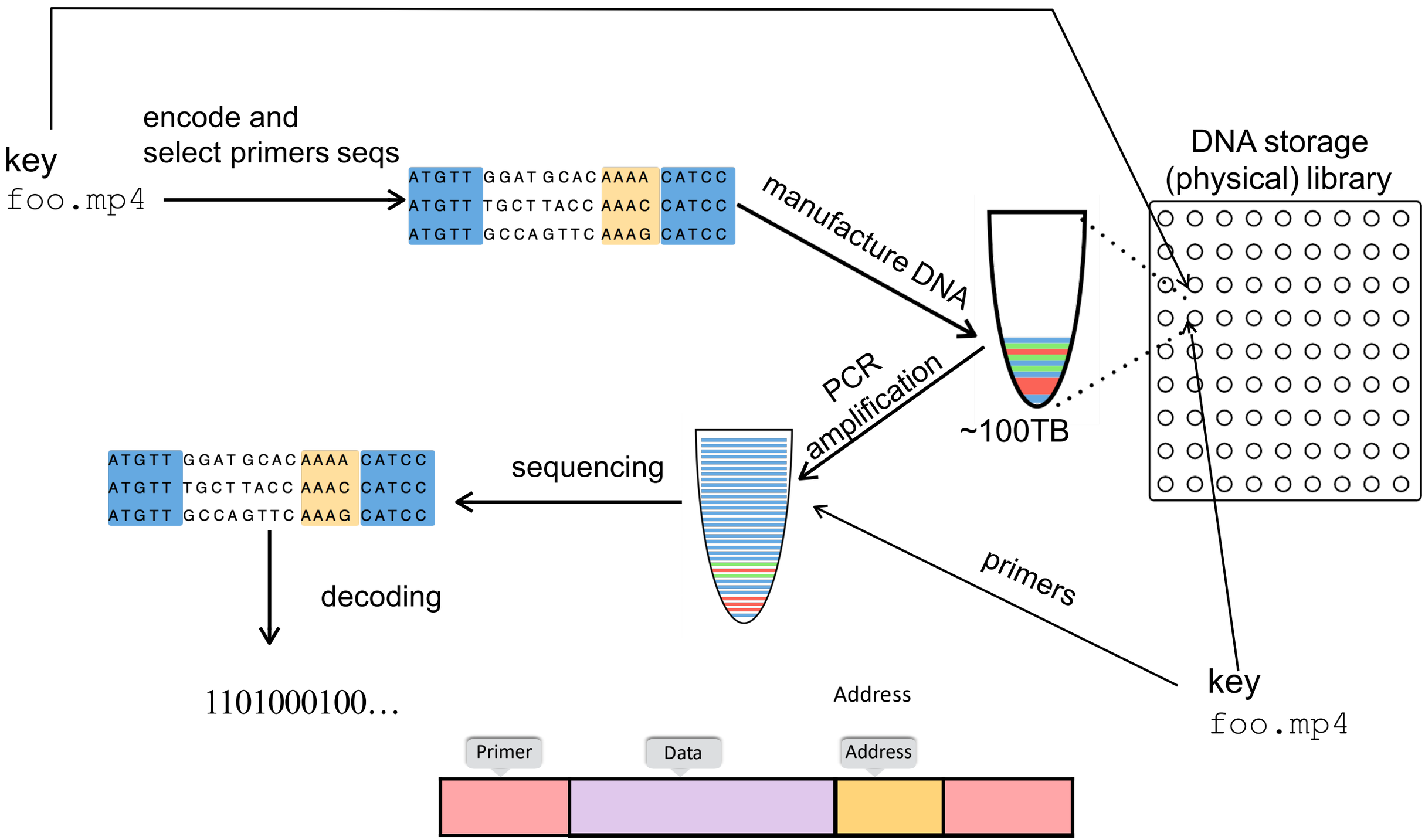
PCR


parent DNA
Step 1: denaturing (95 °C)
DNA primers
DNA polymerase
Taq
nucleotides (dTTP, dCTP, dATP, dGTP)
DNA template strand
Step 2: annealing (55 °C)
two DNA strands
section of DNA to be amplified
Step 3: synthesizing (72 °C)
repeat cycle (20–40 times)
new DNA strands
four DNA strands
© Encyclopædia Britannica, Inc.


Strands with 3 different primers Red, Blue, Green
PCR
Reads are destructive, so replenish when necessary
Selectively amplify strands based on their primer
Sample
Almost all strands have desired primer

Method & system flow

Data address/key specifies physical location and primer for random access.



Binary <-> Ternary <-> nucleotides

3 types of errors:
- substitutions
- Deletion
- insertion

Binary To Ternary: Huffman Encoding with base 3

2222222222112 12011010101010 122111121

2: A
1: T
0: C

AAAAAAAAAAAATTA TACTTCTCTCTCTC TAATTTTAT

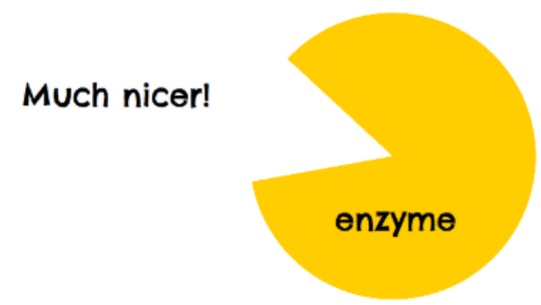
Ternary <-> nucleotides

2222222222112 12011010101010 122111121

ATGCATGCATCTG ATAGACTAGTCGAC TGCTCTCAG

Previous Nucleotide

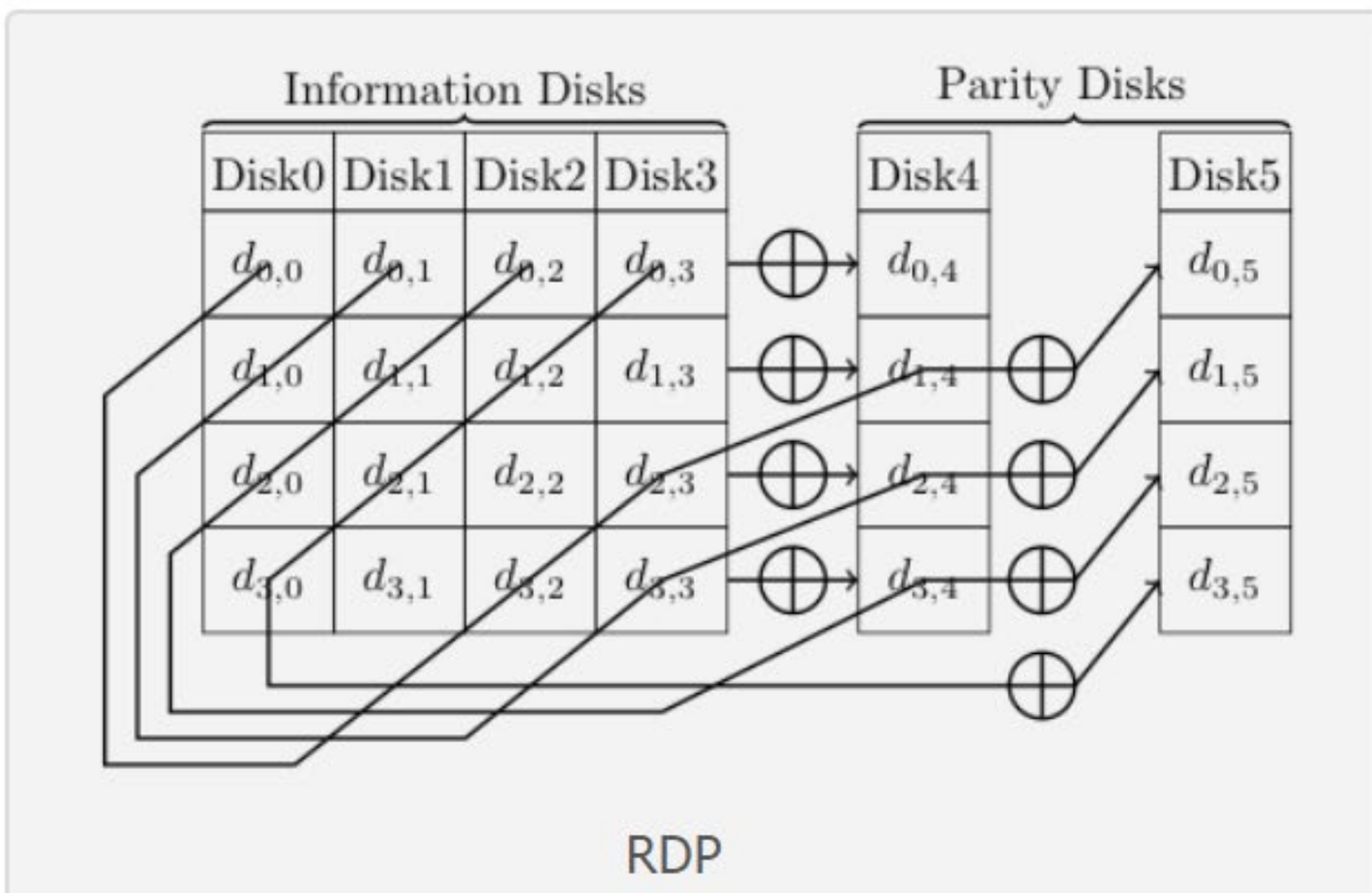
Ternary Digit To Encode	A	C	G	T
0	C	G	T	A
1	G	T	A	C
2	T	A	C	G



Increasing Reliability : RAID reconstruction algorithm

A multi-dimensional algorithm is one in which parts of multiple logical RAID stripes may contribute to parity calculation. Row-Diagonal Parity (RDP) conveyed through diagrams. as in corbett

ATGCATGCATCTGATAGACTAGTCGACTGCTCTCAG



ATGCATGCATCTGATAGA
⊕
CTAGTCGACTGCTCTCAG
⬇
CAGTTGACCATGCCACGG

As long as you get 2 out of 3, you're good to go!

Single Disk Recover

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['GCGAGGCGAAGCCGTCAGAA']	['GCCATTATTTACGGTGAGCA']	['GAAGATAGTAACCTCGCTAC']	['CCCAGATATTGAGTATCCTT']	['TCGGTGGTTAACAACGAGGG']	['TACGAACTGCCATACATACC']
['CCTATTTGTGCCAAAACGGT']	['CCTCCATCGTTAATCGCGAC']	['ACATGCTAACCACTCGGACA']	['TTACCGTATAGAACTTTCAT']	['TGATCAATGACCCCTTCCTC']	['CTACTATCATGGGACATATA']
['TTAAAGCATTTGTTCCTGCA']	['CTTATTAATACCAATGCCTC']	['TAGTGTGTGAAAAGAAATCT']	['ACGATGCGAACCACTAACGG']	['CCTTGAGCGTTGTACTGCCA']	['AGTGGTTGCAGCTTTTTAGG']
['GTTCGGGAGTCCTGGATGAA']	['TTTTAAAATTAGGGCCCGGT']	['GGCCAACAACTCTGGGGGAG']	['TGCGCAGCGCAATGTCACCT']	['AAACTGCCTAGGCAGGATTG']	['GACGCAGAGCTAAGATCGCG']

disk 2 failed

⬇

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['GCGAGGCGAAGCCGTCAGAA']	['GCCATTATTTACGGTGAGCA']	['GAAGATAGTAACCTCGCTAC']	['CCCAGATATTGAGTATCCTT']	['TCGGTGGTTAACAACGAGGG']	['TACGAACTGCCATACATACC']
['CCTATTTGTGCCAAAACGGT']	['CCTCCATCGTTAATCGCGAC']	['ERROR']	['TTACCGTATAGAACTTTCAT']	['TGATCAATGACCCCTTCCTC']	['CTACTATCATGGGACATATA']
['TTAAAGCATTTGTTCCTGCA']	['CTTATTAATACCAATGCCTC']	['TAGTGTGTGAAAAGAAATCT']	['ACGATGCGAACCACTAACGG']	['CCTTGAGCGTTGTACTGCCA']	['AGTGGTTGCAGCTTTTTAGG']
['GTTCGGGAGTCCTGGATGAA']	['TTTTAAAATTAGGGCCCGGT']	['GGCCAACAACTCTGGGGGAG']	['TGCGCAGCGCAATGTCACCT']	['AAACTGCCTAGGCAGGATTG']	['GACGCAGAGCTAAGATCGCG']

Single Disk Recover

disk 2 recovering using row parity

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['GCGAGGCGAAGCCGTCAGAA']	['GCCATTATTTACGGTGAGCA']	['GAAGATAGTAACCTCGCTAC']	['CCCAGATATTGAGTATCCTT']	['TCGGTGGTTAACAACGAGGG']	['TACGAACTGCCATACATACC']
['CCTATTTGTGCCAAAACGGT']	['CCTCCATCGTTAATCGCGAC']	['ERROR']	['TTACCGTATAGAACTTTCAT']	['TGATCAATGACCCCTTCCTC']	['CTACTATCATGGGACATATA']
['TTAAAGCATTTGTTCCTGCA']	['CTTATTAATACCAATGCCTC']	['TAGTGTGTGAAAAGAAATCT']	['ACGATGCGAACCACTAACGG']	['CCTTGAGCGTTGTACTGCCA']	['AGTGGTTGCAGCTTTTTAGG']
['GTTCGGGAGTCCTGGATGAA']	['TTTTAAAATTAGGGCCCGGT']	['GGCCAACAACTCTGGGGGAG']	['TGCGCAGCGCAATGTCACCT']	['AAACTGCCTAGGCAGGATTG']	['GACGCAGAGCTAAGATCGCG']

⊕

['CCTATTTGTGCCAAAACGGT'] ⊕ ['CCTCCATCGTTAATCGCGAC'] ⊕ ['TTACCGTATAGAACTTTCAT'] ⊕ ['TGATCAATGACCCCTTCCTC']

⬇

['ACATGCTAACCACTCGGACA']

writing back the recovered block

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['GCGAGGCGAAGCCGTCAGAA']	['GCCATTATTTACGGTGAGCA']	['GAAGATAGTAACCTCGCTAC']	['CCCAGATATTGAGTATCCTT']	['TCGGTGGTTAACAACGAGGG']	['TACGAACTGCCATACATACC']
['CCTATTTGTGCCAAAACGGT']	['CCTCCATCGTTAATCGCGAC']	['ERROR']	['TTACCGTATAGAACTTTCAT']	['TGATCAATGACCCCTTCCTC']	['CTACTATCATGGGACATATA']
['TTAAAGCATTTGTTCCTGCA']	['CTTATTAATACCAATGCCTC']	['TAGTGTGTGAAAAGAAATCT']	['ACGATGCGAACCACTAACGG']	['CCTTGAGCGTTGTACTGCCA']	['AGTGGTTGCAGCTTTTTAGG']
['GTTCGGGAGTCCTGGATGAA']	['TTTTAAAATTAGGGCCCGGT']	['GGCCAACAACTCTGGGGGAG']	['TGCGCAGCGCAATGTCACCT']	['AAACTGCCTAGGCAGGATTG']	['GACGCAGAGCTAAGATCGCG']

⬇

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['GCGAGGCGAAGCCGTCAGAA']	['GCCATTATTTACGGTGAGCA']	['GAAGATAGTAACCTCGCTAC']	['CCCAGATATTGAGTATCCTT']	['TCGGTGGTTAACAACGAGGG']	['TACGAACTGCCATACATACC']
['CCTATTTGTGCCAAAACGGT']	['CCTCCATCGTTAATCGCGAC']	['ACATGCTAACCACTCGGACA']	['TTACCGTATAGAACTTTCAT']	['TGATCAATGACCCCTTCCTC']	['CTACTATCATGGGACATATA']
['TTAAAGCATTTGTTCCTGCA']	['CTTATTAATACCAATGCCTC']	['TAGTGTGTGAAAAGAAATCT']	['ACGATGCGAACCACTAACGG']	['CCTTGAGCGTTGTACTGCCA']	['AGTGGTTGCAGCTTTTTAGG']
['GTTCGGGAGTCCTGGATGAA']	['TTTTAAAATTAGGGCCCGGT']	['GGCCAACAACTCTGGGGGAG']	['TGCGCAGCGCAATGTCACCT']	['AAACTGCCTAGGCAGGATTG']	['GACGCAGAGCTAAGATCGCG']

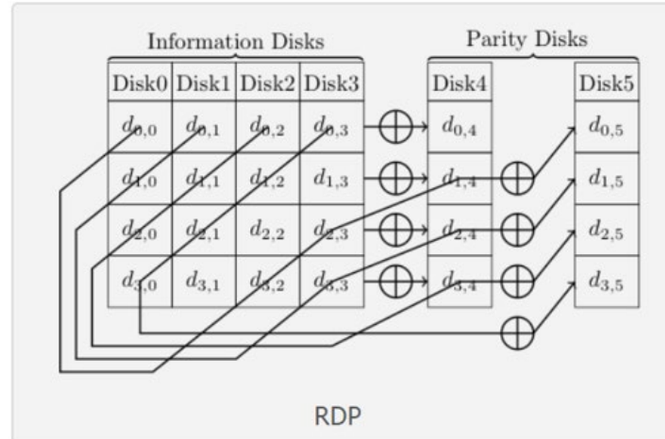
Double Disk Recover

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['TCTGCGCATGCGTCAGTGAG']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['CTCGAGAGCAGCATCACACG']	['ATCATGCGTGACGCTACGAC']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CAGCGTCGTGTACAGCACGC']	['TGATCTAGCTCGATCTAGCT']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['TGCTGTGTATGCGCGTATCA']	['CGATCTAGCTCGATCTAGCT']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

disks 1 and 2 failed

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['ERROR']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

Double Disk Recover



disk 2 block recovering using diagonal parity

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['ERROR']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

['TCATGCACGTACATCTCGAT'] ⊕ ['CGATGCATAGAGCGCATGCA'] ⊕ ['CGTCAGCCCGTTGGGTTGCA'] ⊕ ['CCTTCGCAAACGCGGTGCTC']

⬇

['CGATCTAGCTCGATCTAGCT']

writing back the recovered block

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['ERROR']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

⬇

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['CGATCTAGCTCGATCTAGCT']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

Double Disk Recover

disk 1 recovering using row parity

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['CGATCTAGCTCGATCTAGCT']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

['CAGTGTCAGACAGACTGCGT'] ⊕ ['CGATCTAGCTCGATCTAGCT'] ⊕ ['CATGCAGATCGTACGTGTAC'] ⊕ ['GAACATCCACAAATAAATGC']

⬇

['CGCGGATGAGGCGGGTAATC']

writing back the recovered block

['CAGTGTCAGACAGACTGCGT']	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['ERROR']	['CGATCTAGCTCGATCTAGCT']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

⬇

data disk 0	data disk 1	data disk 2	data disk 3	Row Parity	Diagonal Parity
['TCATGCACGTACATCTCGAT']	['ATACTGATAGCTGTGCATGT']	['GAACAAAATGAAGTGCCTTT']	['CGATCTAGCTCGATCTAGCT']	['CCTTCGCAAACGCGGTGCTC',	'CCTTCGCAAACGCGGTGCTC']
['CATAGCATGTACATCATGCA']	['AGGCCACGCATTCCTTTCGT']	['AACGGGCTGAGACCTAACGA']	['CGATCTAGCTCGATCTAGCT']	['CGTCAGCCCGTTGGGTTGCA',	'CGTCAGCCCGTTGGGTTGCA']
['CACTACTACAGTCGCGTATG']	['CTGTTTCTGACCGAGCCGAA']	['CCTTCTGATGTAGAGTCAGG']	['CGATGCATAGAGCGCATGCA']	['GATGCAGTTTATCTTAACCA',	'GATGCAGTTTATCTTAACCA']
['CAGTGTCAGACAGACTGCGT']	['CGCGGATGAGGCGGGTAATC']	['CGATCTAGCTCGATCTAGCT']	['CATGCAGATCGTACGTGTAC']	['GAACATCCACAAATAAATGC',	'GAACATCCACAAATAAATGC']

Software Implementation – Python

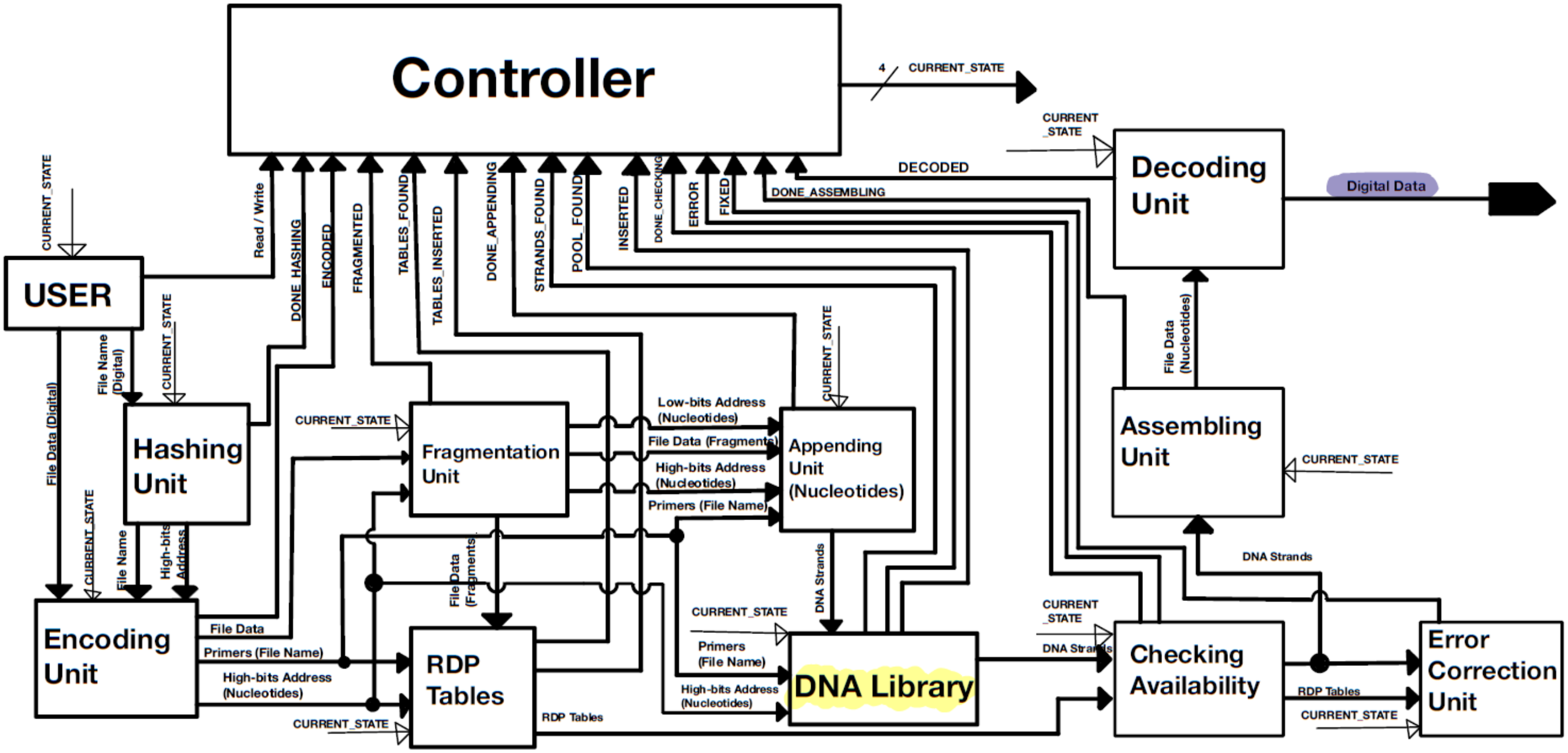
We implemented the algorithm for transforming binary data into DNA in Python, which you can learn more about here: [github/DNA-Storage-System]

```
GATACAGTCAAAGTGTCGAGTGAGAGATCACCTGCGATGCATGCAGTATCGAGACGTGCGCA
CTATGTCAGTTTCACAGCTCACTCTCTAGTGGACGCTACGTACGTCATAGCTCTGCACGCGT

GATACAGTCAATGTGACTGATGCATAGAGTGCTGCGATGCATGCAGTATCTAGACGTGCGCA
CTATGTCAGTTACACTGACTACGTATCTCACGACGCTACGTACGTCATAGATCTGCACGCGT

GATACAGTCAACTCTCAGCGCTGATATAGATCTGCGATGCATGCAGTATCAAGACGTGCGCA
CTATGTCAGTTGAGAGTCGCGACTATATCTAGACGCTACGTACGTCATAGTTCTGCACGCGT
```

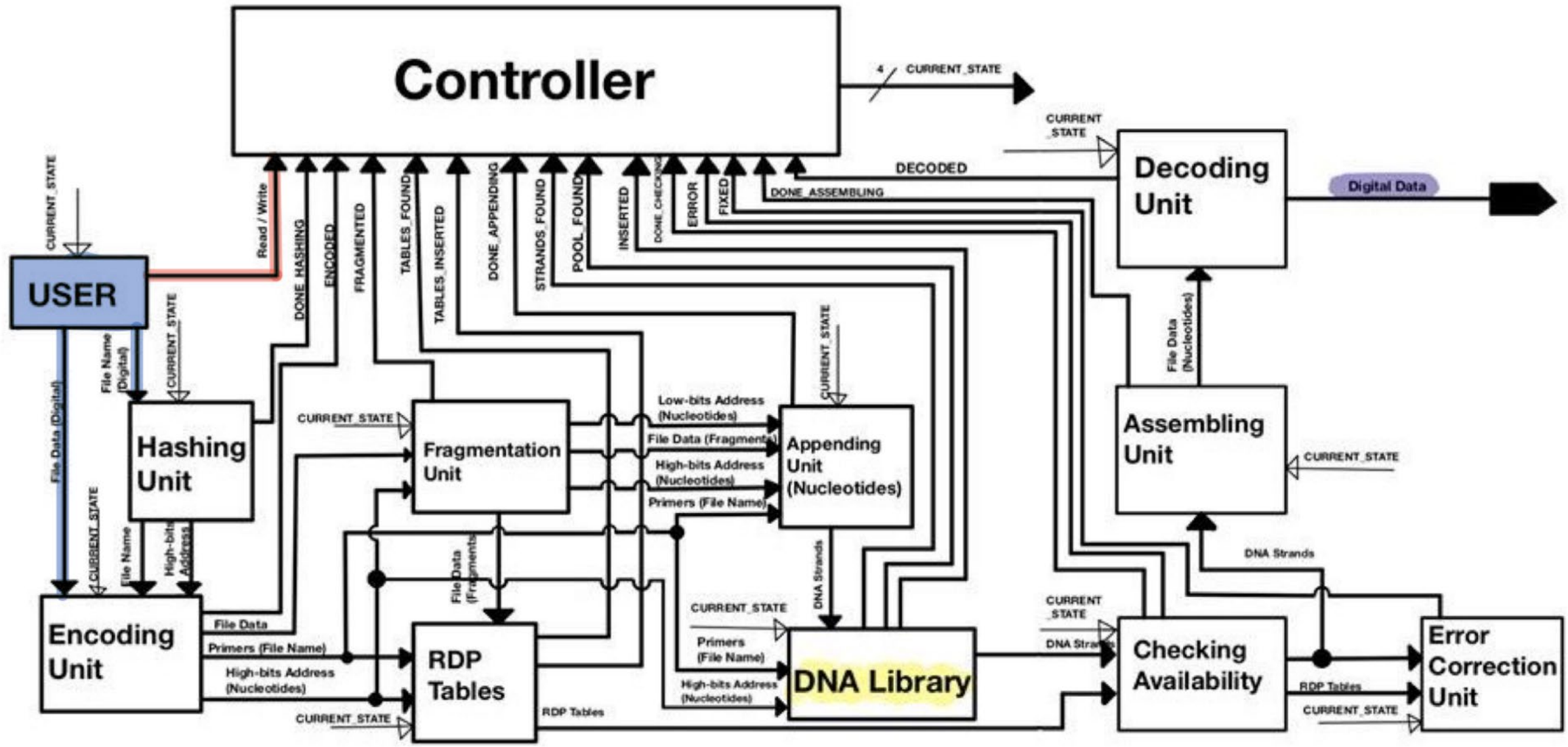
```
                                    Diagonal parity reocover
       data disk 0              data disk 1              data disk 2              data disk 3              Row Parity               Diagonal Parity
['TTCAGAAATAAACACTGGGC'] ['CATTCAGCTGTTCAGGGTCC'] ['TTAGACATCCCTCCCGTGGC'] ['TTTCGAAACGCCAACCGTGA'] ['GTCACCGGACTCCCTGCATC'] ['GAAATCAACTGACCAACTCG']
['TTGACATGCGCACATGGTTG'] ['TGTTCCCATCTACACCTTAT']         ['ERROR']        ['CACTTCATCGACGAGTGGTC'] ['TGCATCGAAGATAATCGCAG'] ['AGCACTGCAGCCCGTTAGTC']
['ACCCGGGTTCCCGGGGTTCA'] ['AAGAAAATTGTCTTGGAAGT'] ['TGCCGGGCGGGCAATTGCGA'] ['TTCGTTGGCTCGTCATGTGT'] ['AATGTTGTTGCTGATATCTA'] ['TGTGATACTGAATTCAACAT']
['ATAAGACGGTGACCAAATCG'] ['AGTCGGTAGGAGATATTACT'] ['CGCGCTAGTGTCTAGCTTCG'] ['GCTCGACCCAGCTGCTTTGG'] ['TGCGTCTCGTTGCATCTTTC'] ['CTACACCATAGGAGAGTTCT']
       data disk 0              data disk 1              data disk 2              data disk 3              Row Parity               Diagonal Parity
['TTCAGAAATAAACACTGGGC'] ['CATTCAGCTGTTCAGGGTCC'] ['TTAGACATCCCTCCCGTGGC'] ['TTTCGAAACGCCAACCGTGA'] ['GTCACCGGACTCCCTGCATC'] ['GAAATCAACTGACCAACTCG']
['TTGACATGCGCACATGGTTG'] ['TGTTCCCATCTACACCTTAT'] ['GTCAACACTTGGGATCCTAG'] ['CACTTCATCGACGAGTGGTC'] ['TGCATCGAAGATAATCGCAG'] ['AGCACTGCAGCCCGTTAGTC']
['ACCCGGGTTCCCGGGGTTCA'] ['AAGAAAATTGTCTTGGAAGT'] ['TGCCGGGCGGGCAATTGCGA'] ['TTCGTTGGCTCGTCATGTGT'] ['AATGTTGTTGCTGATATCTA'] ['TGTGATACTGAATTCAACAT']
['ATAAGACGGTGACCAAATCG'] ['AGTCGGTAGGAGATATTACT'] ['CGCGCTAGTGTCTAGCTTCG'] ['GCTCGACCCAGCTGCTTTGG'] ['TGCGTCTCGTTGCATCTTTC'] ['CTACACCATAGGAGAGTTCT']

                                          SUCCESS
```

```
                                       Row parity reocover
       data disk 0              data disk 1              data disk 2              data disk 3              Row Parity               Diagonal Parity
['TTCAGAAATAAACACTGGGC'] ['CATTCAGCTGTTCAGGGTCC'] ['TTAGACATCCCTCCCGTGGC'] ['TTTCGAAACGCCAACCGTGA'] ['GTCACCGGACTCCCTGCATC'] ['GAAATCAACTGACCAACTCG']
['TTGACATGCGCACATGGTTG'] ['TGTTCCCATCTACACCTTAT']         ['ERROR']        ['CACTTCATCGACGAGTGGTC'] ['TGCATCGAAGATAATCGCAG'] ['AGCACTGCAGCCCGTTAGTC']
['ACCCGGGTTCCCGGGGTTCA'] ['AAGAAAATTGTCTTGGAAGT'] ['TGCCGGGCGGGCAATTGCGA'] ['TTCGTTGGCTCGTCATGTGT'] ['AATGTTGTTGCTGATATCTA'] ['TGTGATACTGAATTCAACAT']
['ATAAGACGGTGACCAAATCG'] ['AGTCGGTAGGAGATATTACT'] ['CGCGCTAGTGTCTAGCTTCG'] ['GCTCGACCCAGCTGCTTTGG'] ['TGCGTCTCGTTGCATCTTTC'] ['CTACACCATAGGAGAGTTCT']
       data disk 0              data disk 1              data disk 2              data disk 3              Row Parity               Diagonal Parity
['TTCAGAAATAAACACTGGGC'] ['CATTCAGCTGTTCAGGGTCC'] ['TTAGACATCCCTCCCGTGGC'] ['TTTCGAAACGCCAACCGTGA'] ['GTCACCGGACTCCCTGCATC'] ['GAAATCAACTGACCAACTCG']
['TTGACATGCGCACATGGTTG'] ['TGTTCCCATCTACACCTTAT'] ['GTCAACACTTGGGATCCTAG'] ['CACTTCATCGACGAGTGGTC'] ['TGCATCGAAGATAATCGCAG'] ['AGCACTGCAGCCCGTTAGTC']
['ACCCGGGTTCCCGGGGTTCA'] ['AAGAAAATTGTCTTGGAAGT'] ['TGCCGGGCGGGCAATTGCGA'] ['TTCGTTGGCTCGTCATGTGT'] ['AATGTTGTTGCTGATATCTA'] ['TGTGATACTGAATTCAACAT']
['ATAAGACGGTGACCAAATCG'] ['AGTCGGTAGGAGATATTACT'] ['CGCGCTAGTGTCTAGCTTCG'] ['GCTCGACCCAGCTGCTTTGG'] ['TGCGTCTCGTTGCATCTTTC'] ['CTACACCATAGGAGAGTTCT']

                                          SUCCESS
```

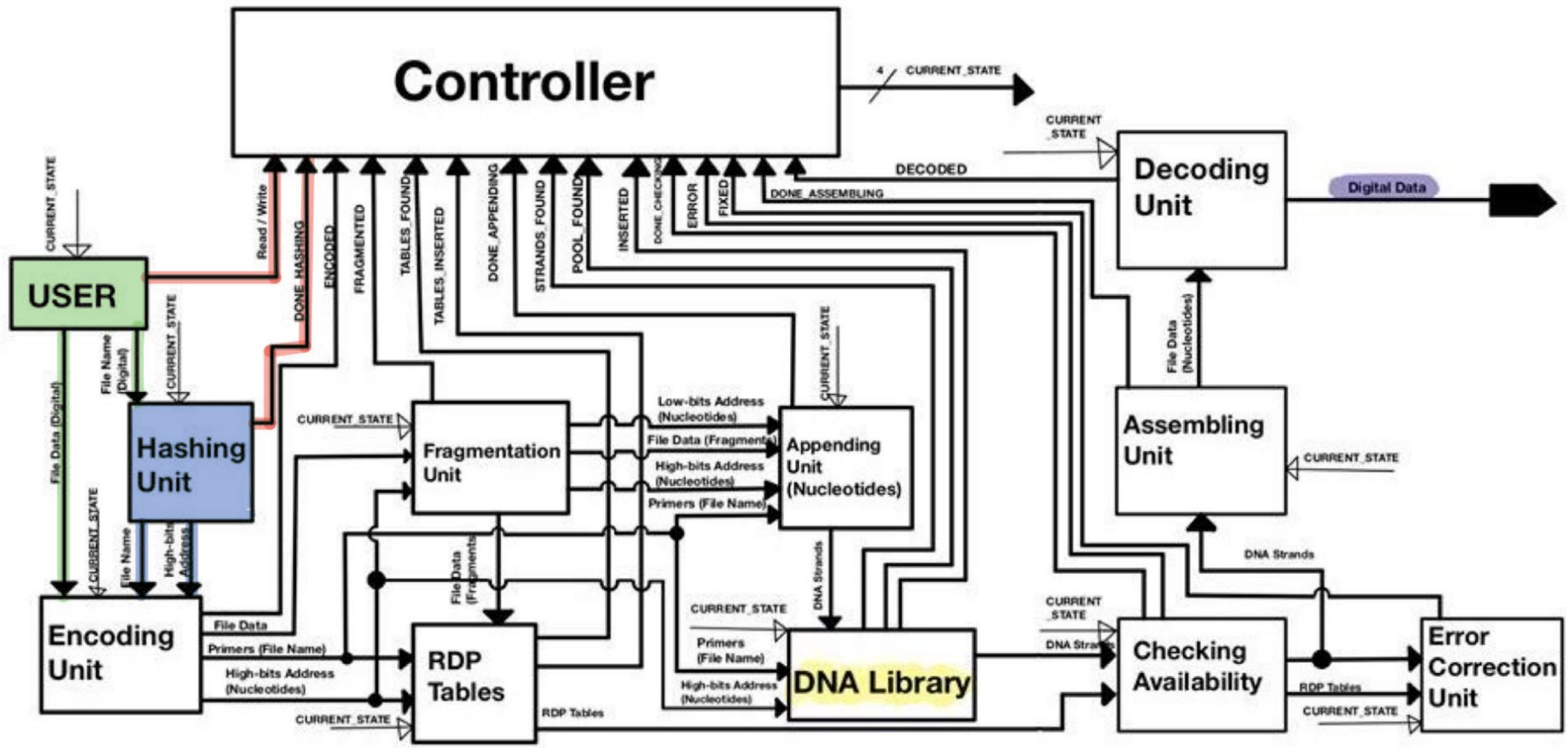
DNA system Controller & data path



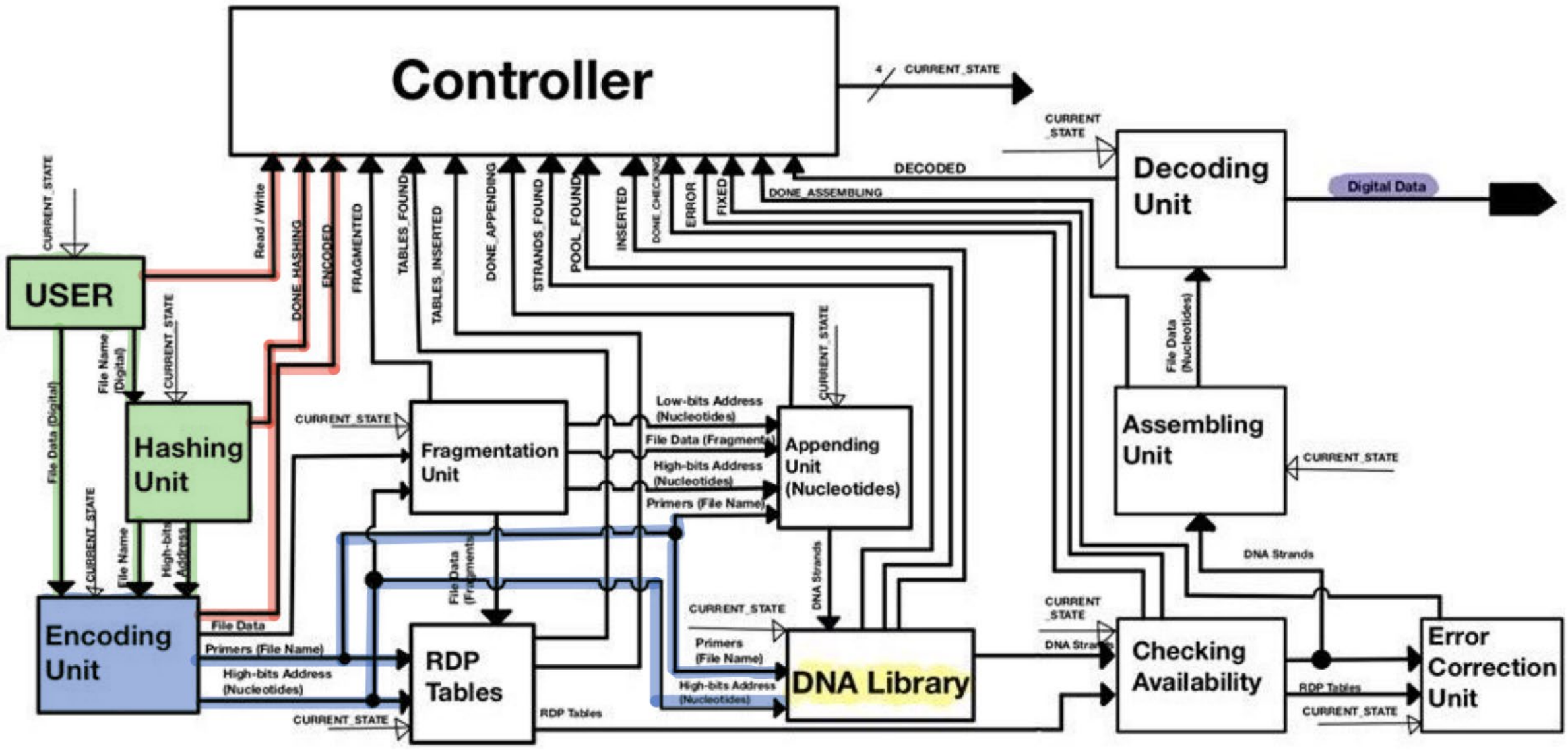
DNA system Controller & data path read simulation



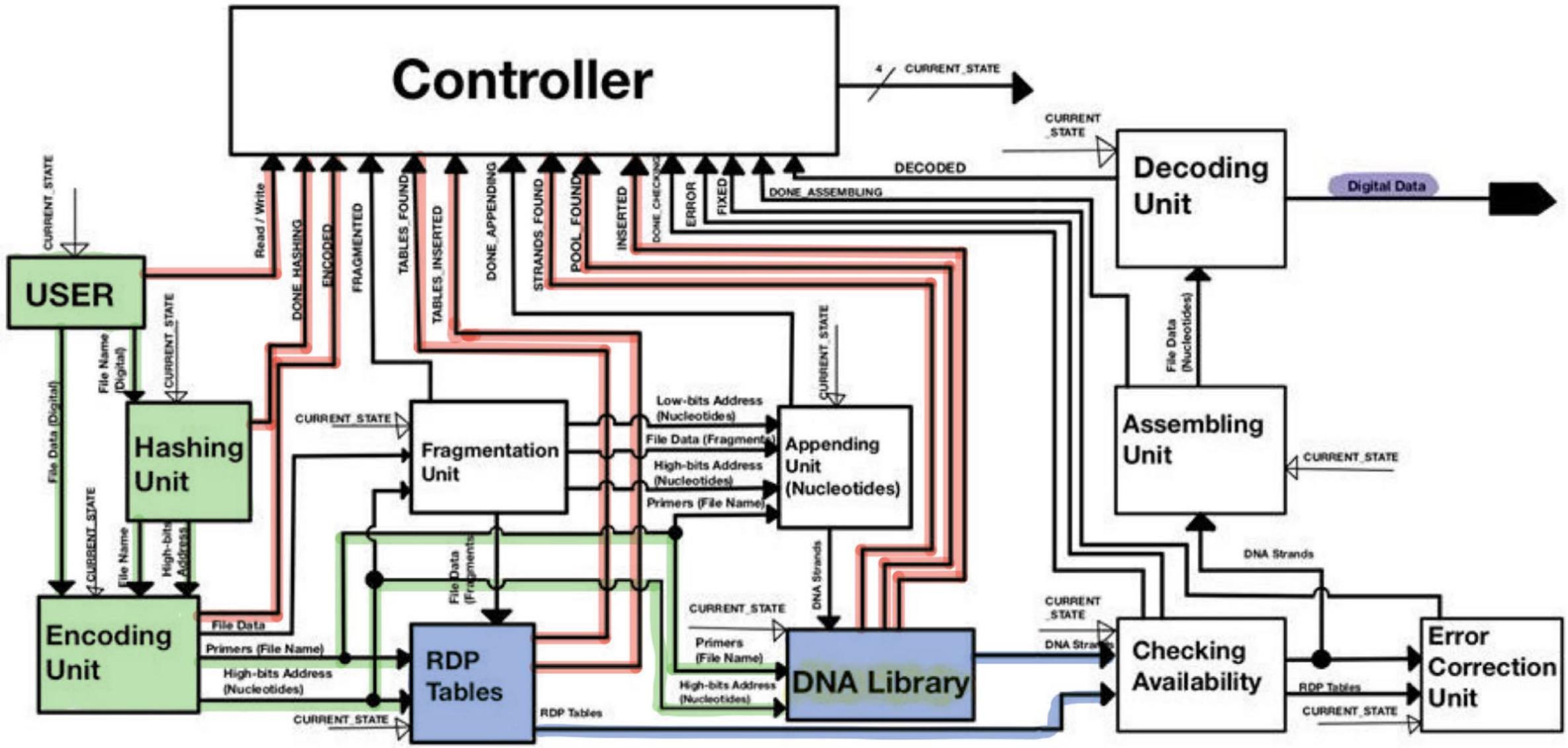
DNA system Controller & data path read simulation



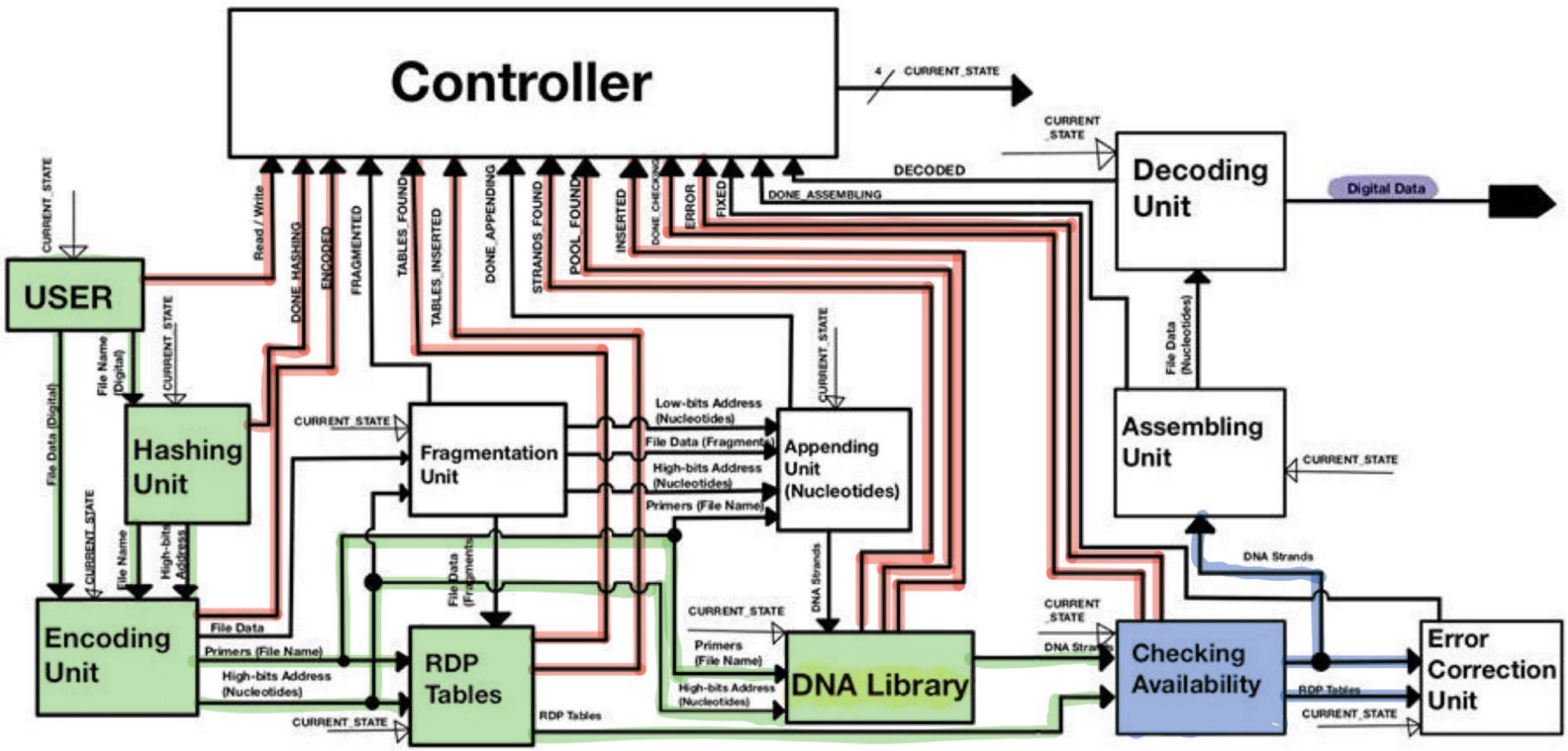
DNA system Controller & data path read simulation



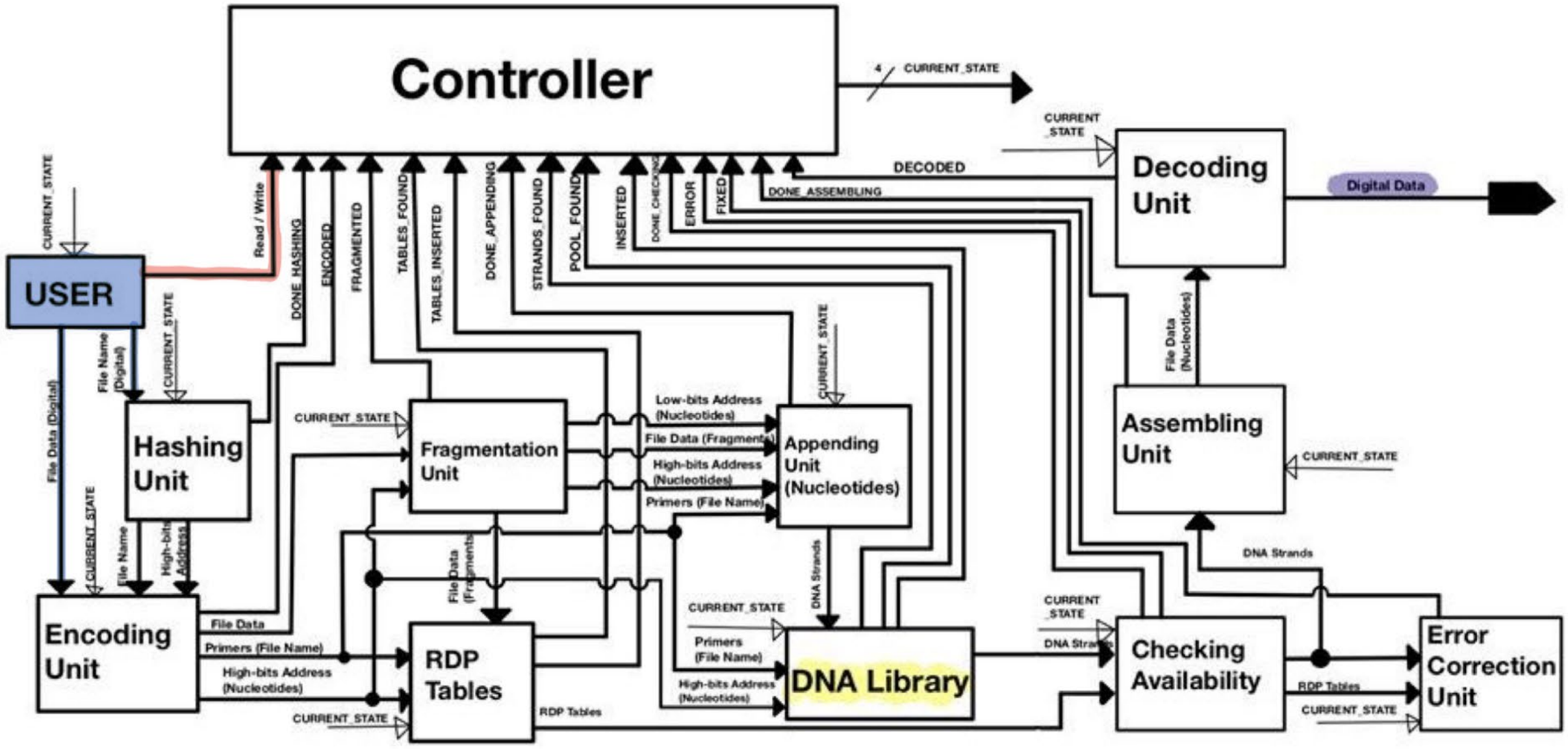
DNA system Controller & data path read simulation



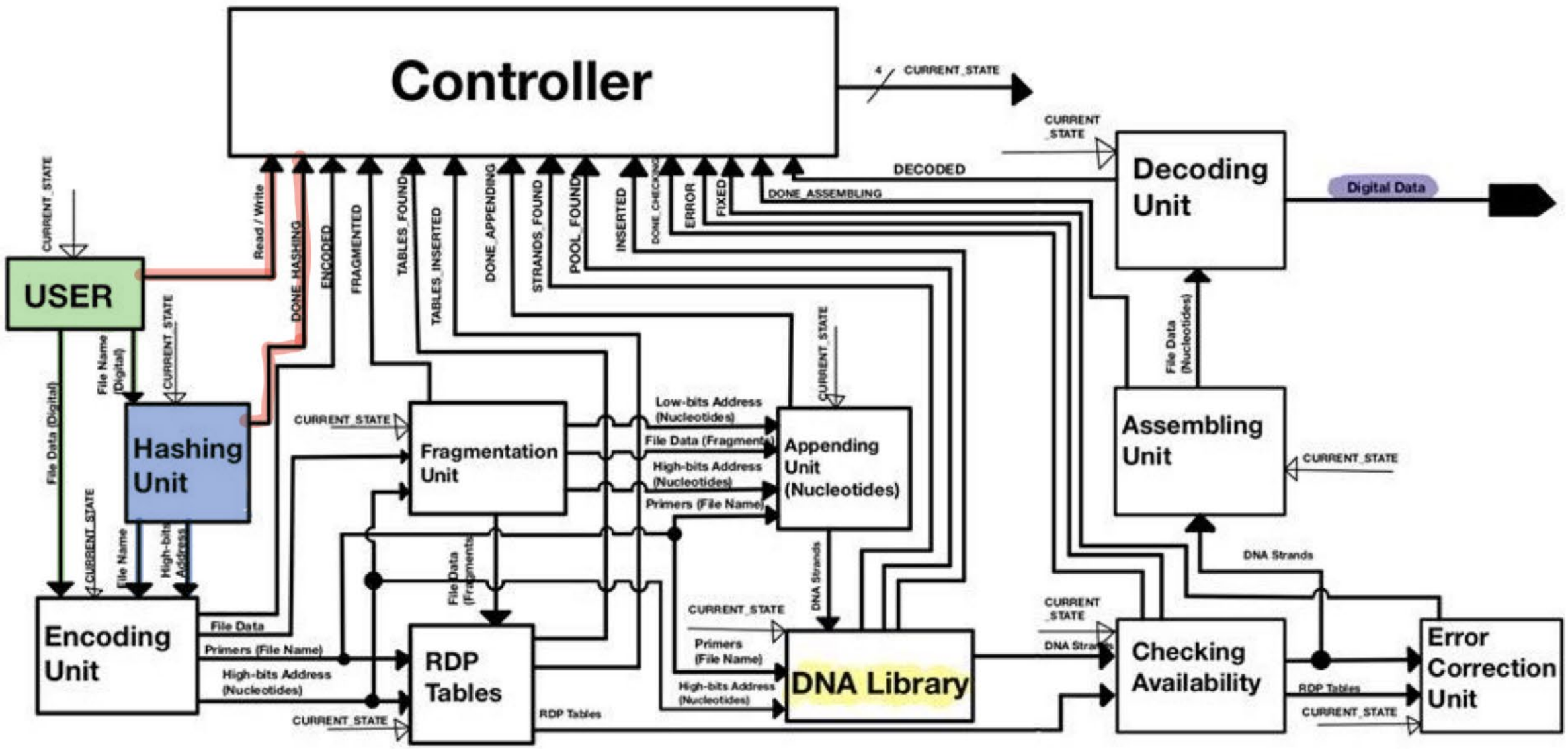
DNA system Controller & data path read simulation



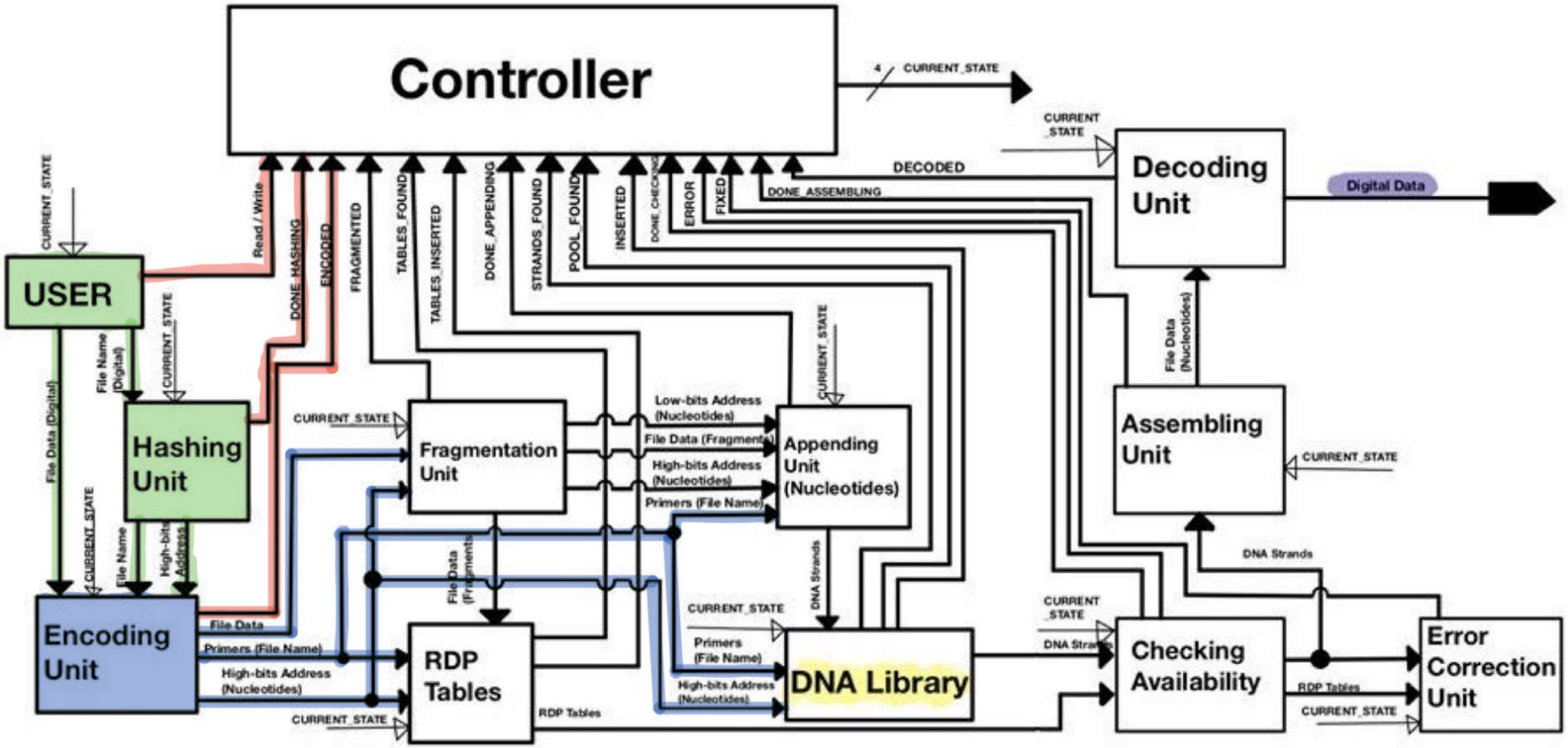
DNA system Controller & data path write simulation



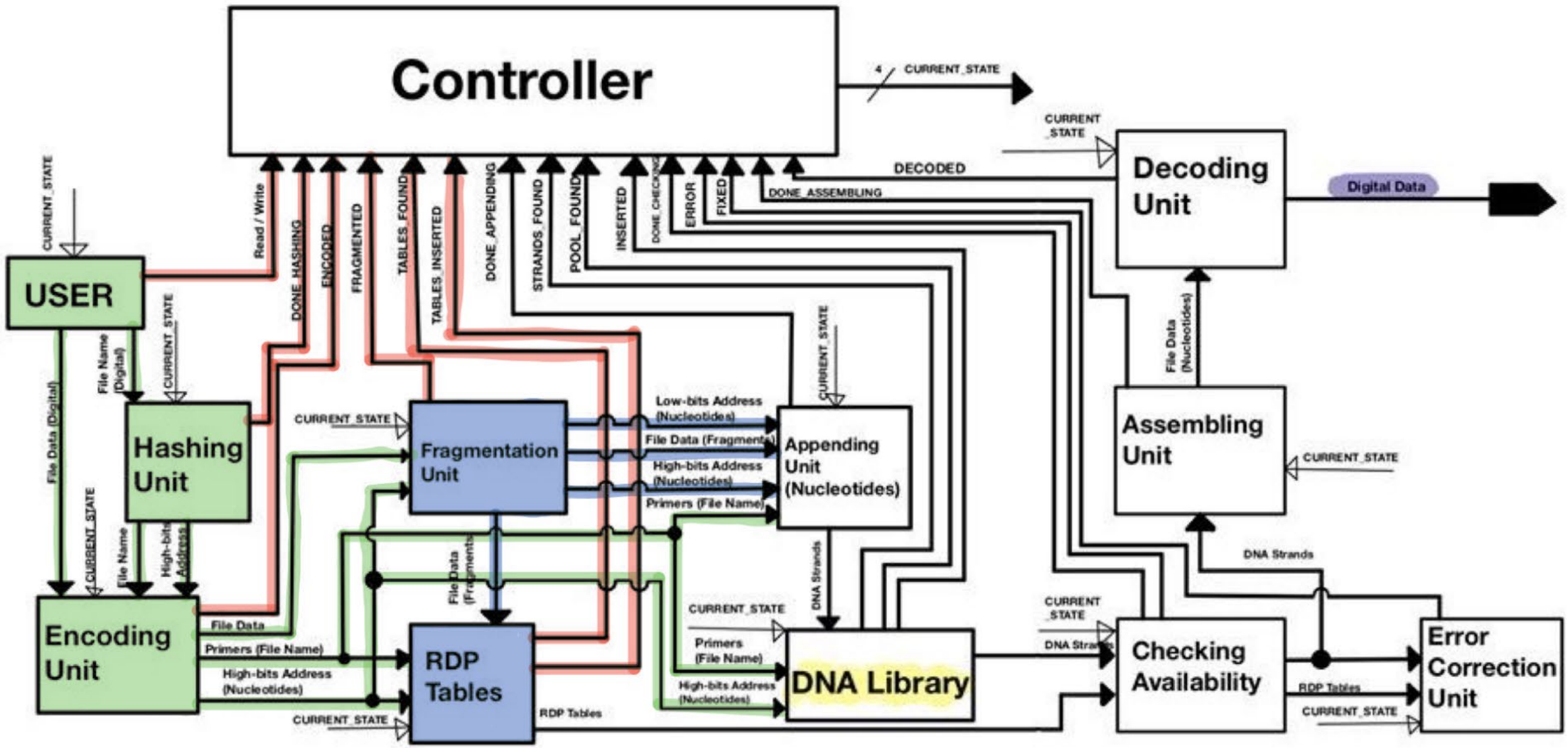
DNA system Controller & data path write simulation



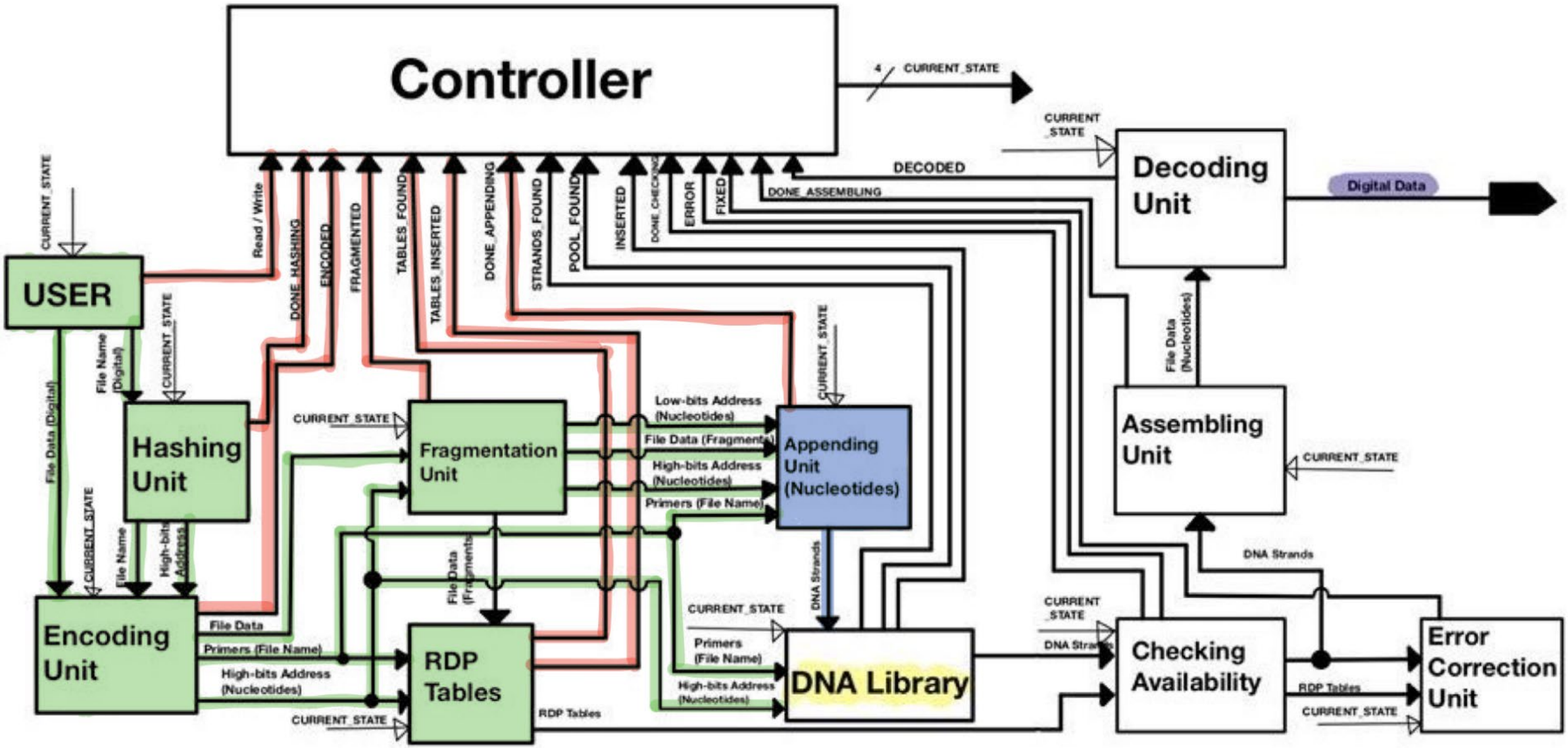
DNA system Controller & data path write simulation



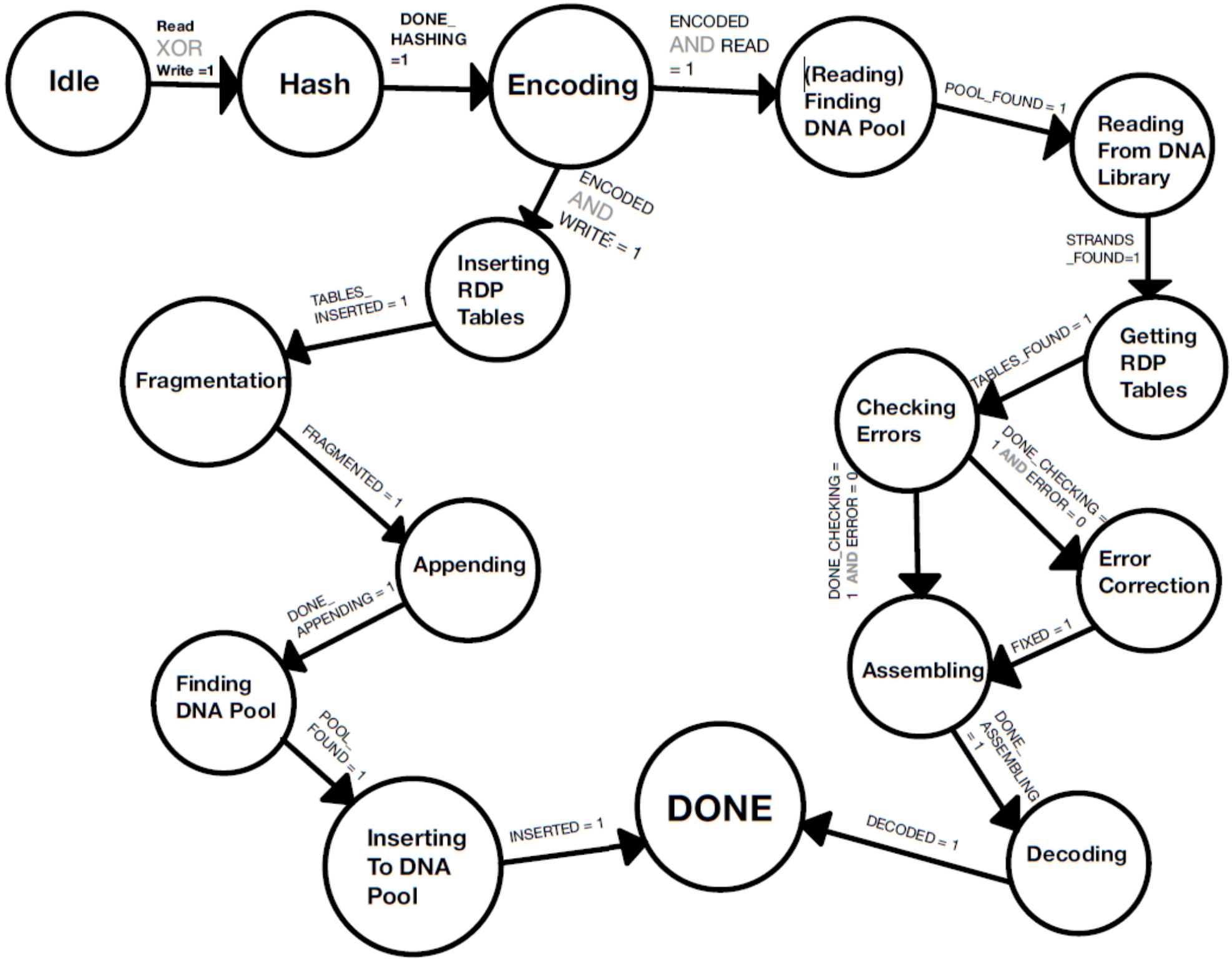
DNA system Controller & data path write simulation



DNA system Controller & data path write simulation



DNA system state machine



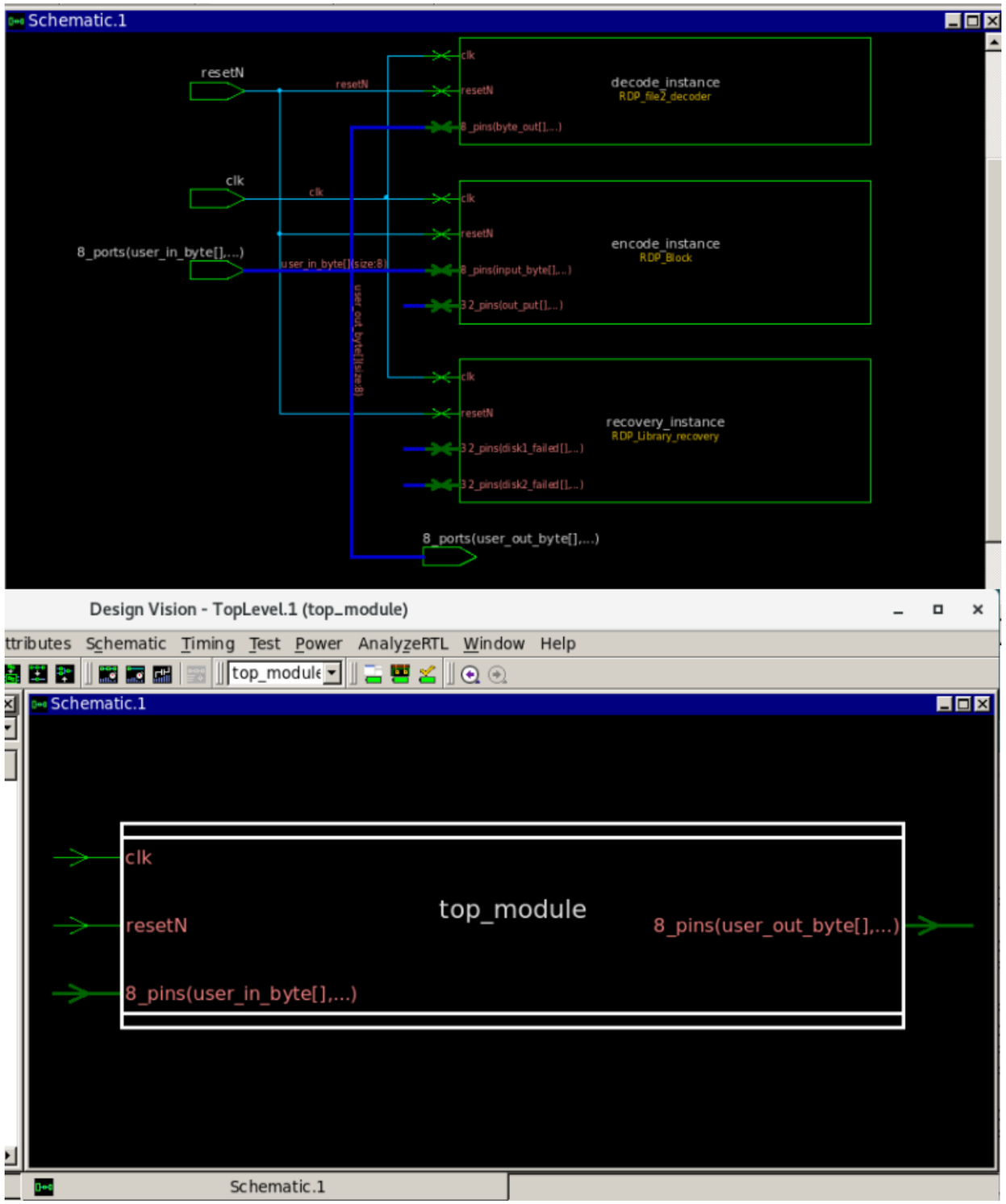
Hardware implementation: system verilog

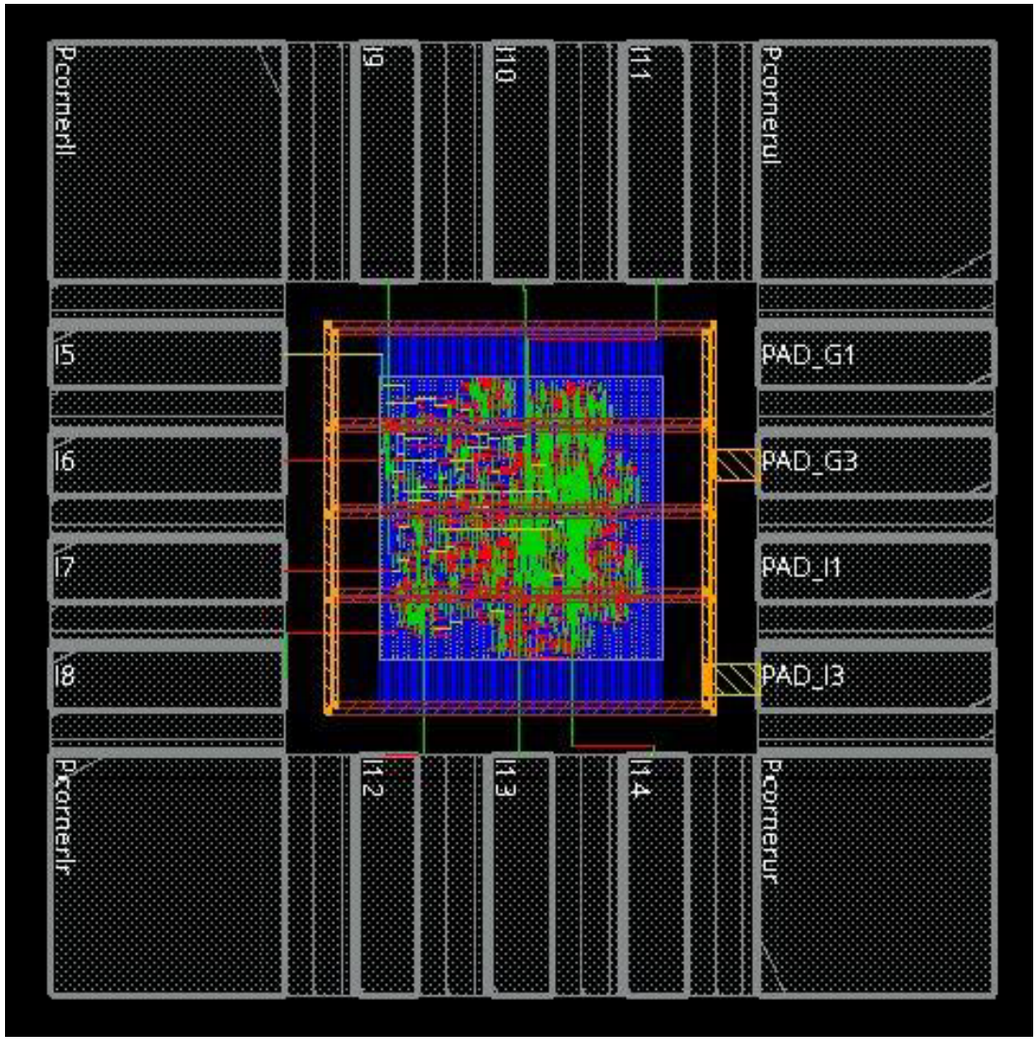
```
30 s
 ----------------  RDP Library ----------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
1000110010000001 | 0111101110000001 | 1101001110000001 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
1110010010000000 | 0110001100000001 | 1110000100000001 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
1110011100000001 | 1110000111000001 | 0011101100000001 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0001000110000001 | 0111001001000001 | 0110001001000001 | 1001100010000001 | 1001100100000000 | 0101110011000000 |


 ----------------  RDP Library, single disk 1 failed ----------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
0000000000000000 | 0111101110000001 | 1101001110000001 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
0000000000000000 | 0110001100000001 | 1110000100000001 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
0000000000000000 | 1110000111000001 | 0011101100000001 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0000000000000000 | 0111001001000001 | 0110001001000001 | 1001100010000001 | 1001100100000000 | 0101110011000000 |


--------- recovery disk number 1
 new D1_1 = 1000110010000001
 new D1_2 = 1110010010000000
 new D1_3 = 1110011100000001
 new D1_4 = 0001000110000001


 ----------------  RDP Library, disk 1 recovery ----------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
1000110010000001 | 0111101110000001 | 1101001110000001 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
1110010010000000 | 0110001100000001 | 1110000100000001 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
1110011100000001 | 1110000111000001 | 0011101100000001 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0001000110000001 | 0111001001000001 | 0110001001000001 | 1001100010000001 | 1001100100000000 | 0101110011000000 |
```

Hardware implementation: system verilog

```
30 s
------------------ RDP Library ------------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
1000110010000001 | 0111101110000001 | 1101001110000001 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
1110010010000000 | 0110001100000001 | 1110000100000001 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
1110011100000001 | 1110000111000001 | 0011101100000001 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0001000110000001 | 0111001001000001 | 0110001001000001 | 1001100010000001 | 1001100100000000 | 0101110011000000 |


------------------ RDP Library, double_disk_recover ------------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
0000000000000000 | 0111101110000001 | 0000000000000000 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
0000000000000000 | 0110001100000001 | 0000000000000000 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
0000000000000000 | 1110000111000001 | 0000000000000000 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0000000000000000 | 0111001001000001 | 0000000000000000 | 1001100010000001 | 1001100100000000 | 0101110011000000 |

--------- recovery disks [ 1 , 3 ]


------------------ RDP Library, disks 1 , 3 recovery ------------------
     disk 1      |      disk 2      |      disk 3      |      disk 4      |    row Parity    |    diag Parity   |
1000110010000001 | 0111101110000001 | 1101001110000001 | 0100110110000001 | 0110100100000000 | 1101100100000000 |
1110010010000000 | 0110001100000001 | 1110000100000001 | 0111011011000001 | 0001000001000001 | 0001110111000000 |
1110011100000001 | 1110000111000001 | 0011101100000001 | 0010011110000001 | 0001101001000000 | 1100111010000001 |
0001000110000001 | 0111001001000001 | 0110001001000001 | 1001100010000001 | 1001100100000000 | 0101110011000000 |
```

Synthesis and Layout





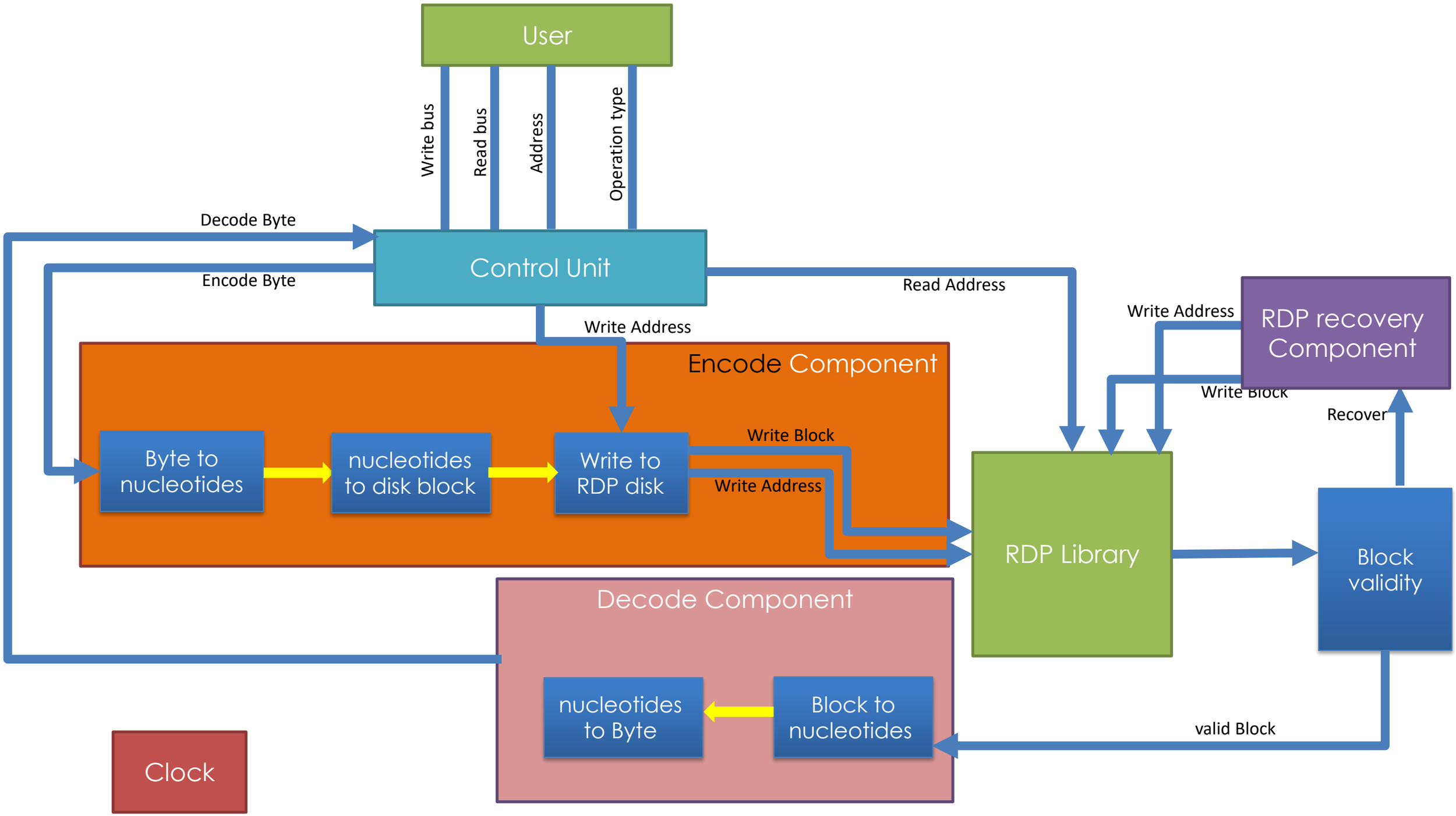
Conclusion

- Better error correction
- Better random access
- Software > Hardware

Future work

- Faster sequencing
- Higher level of resolution

DNA system data path, after implementation



The system in a nutshell

System overview

Archival storage system structured as a key-value store

System overview

Archival storage system structured as a key-value store

put(key, value)

System overview

Archival storage system structured as a key-value store

get(key)

System overview

Archival storage system structured as a key-value store

put(key, value)

get(key)

System overview

Archival storage system structured as a key-value store

cat.jpg

01001…

put(key, value)

get(key)

System overview

Archival storage system structured as a key-value store

cat.jpg

01001…

put(key, value)

ACGAT…

get(key)

System overview

Archival storage system structured as a key-value store

cat.jpg

01001…

put(key, value)

ACGAT…

get(key)

System overview

Archival storage system structured as a key-value store

cat.jpg

01001…

put(key, value)

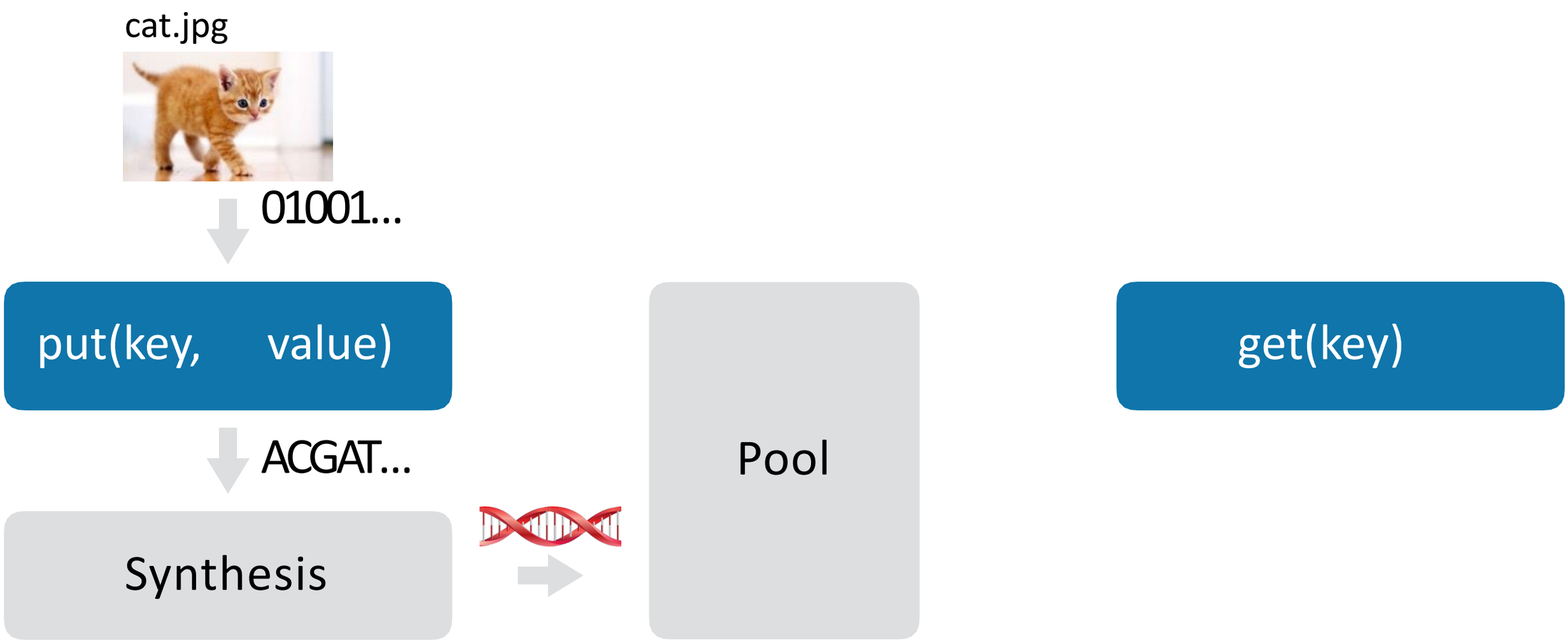
get(key)

ACGAT…

Synthesis

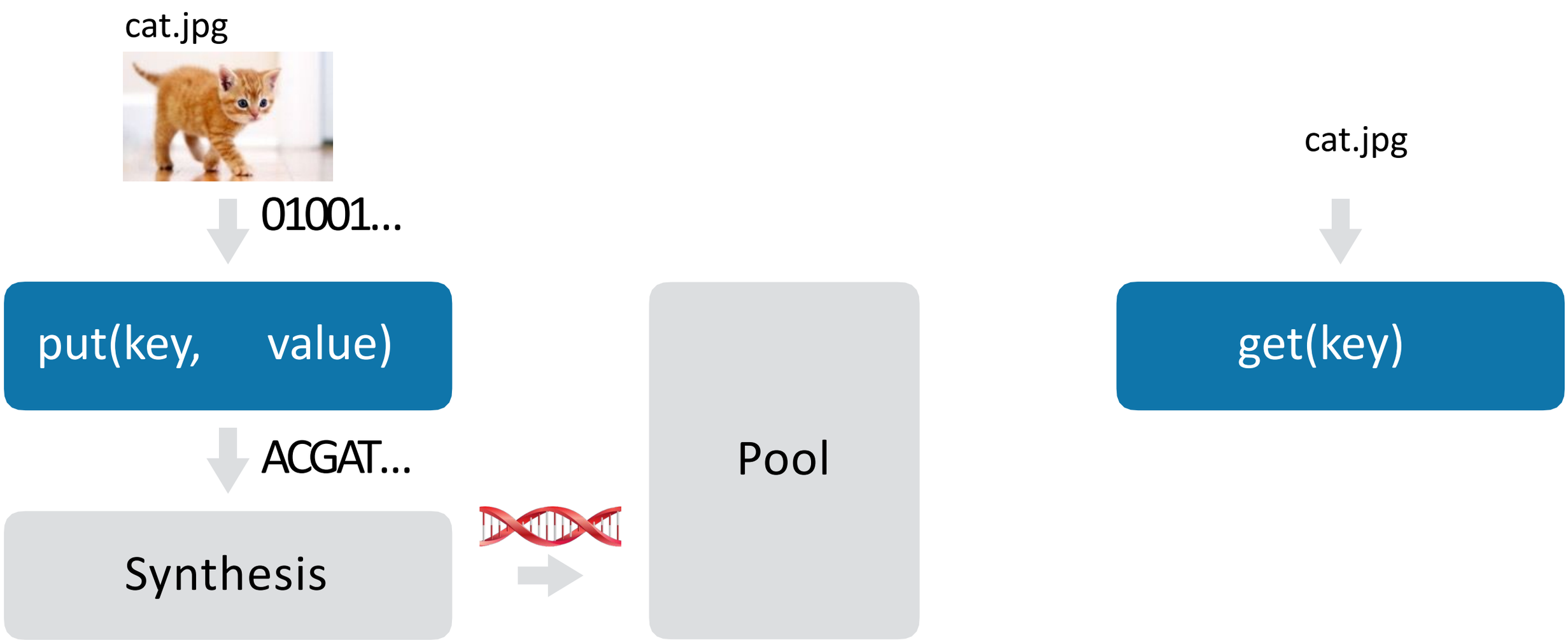
System overview

Archival storage system structured as a key-value store



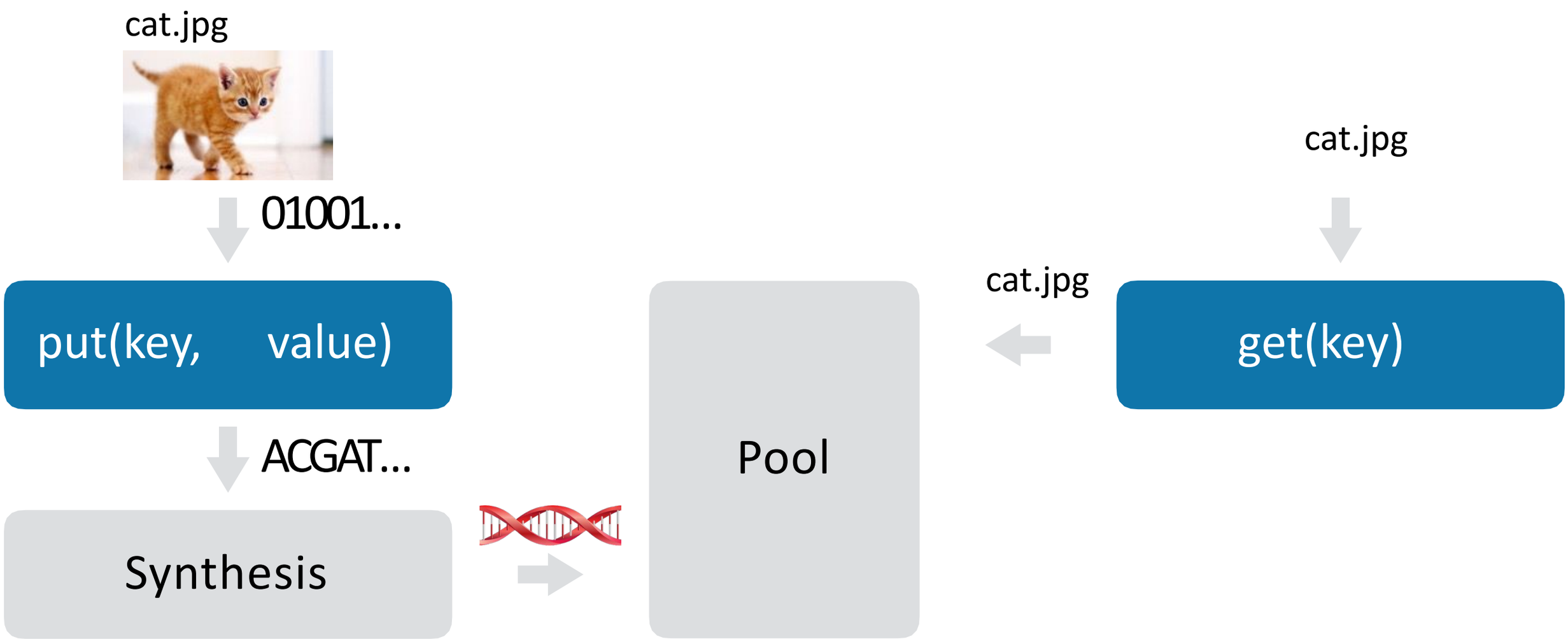
System overview

Archival storage system structured as a key-value store



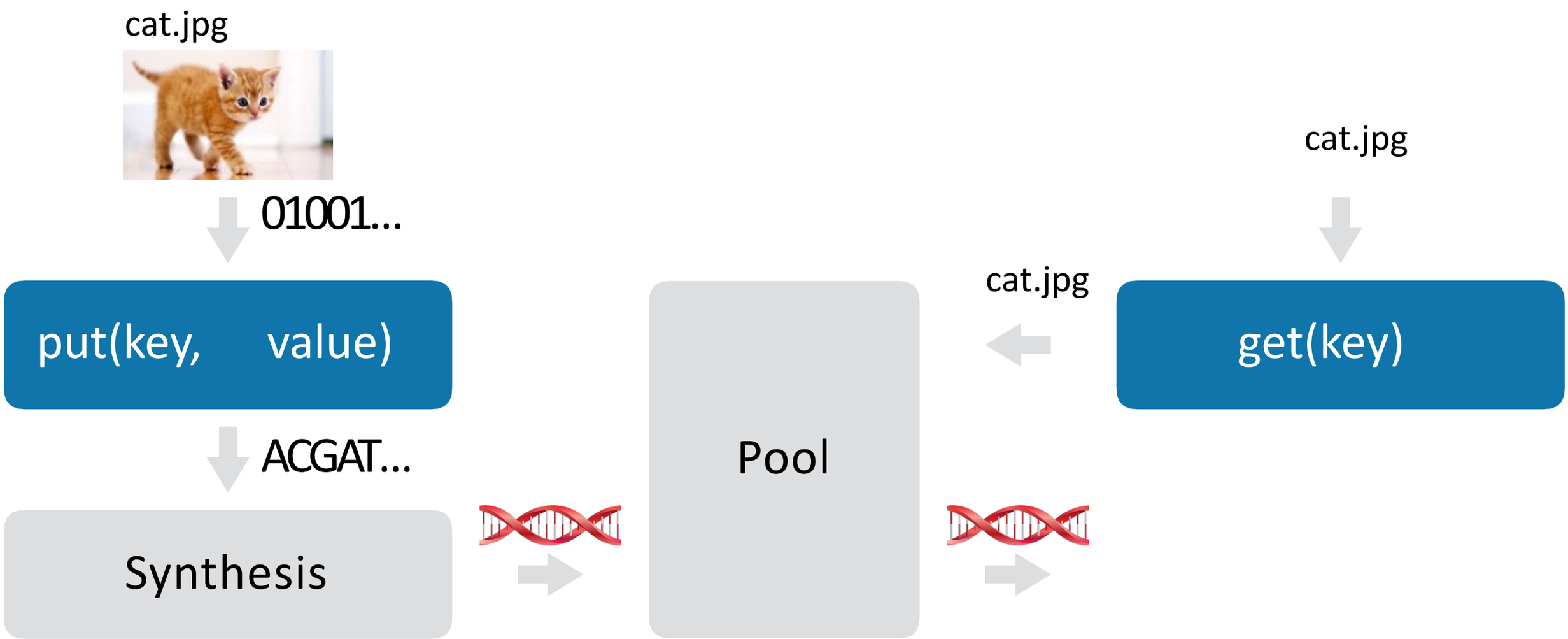
System overview

Archival storage system structured as a key-value store



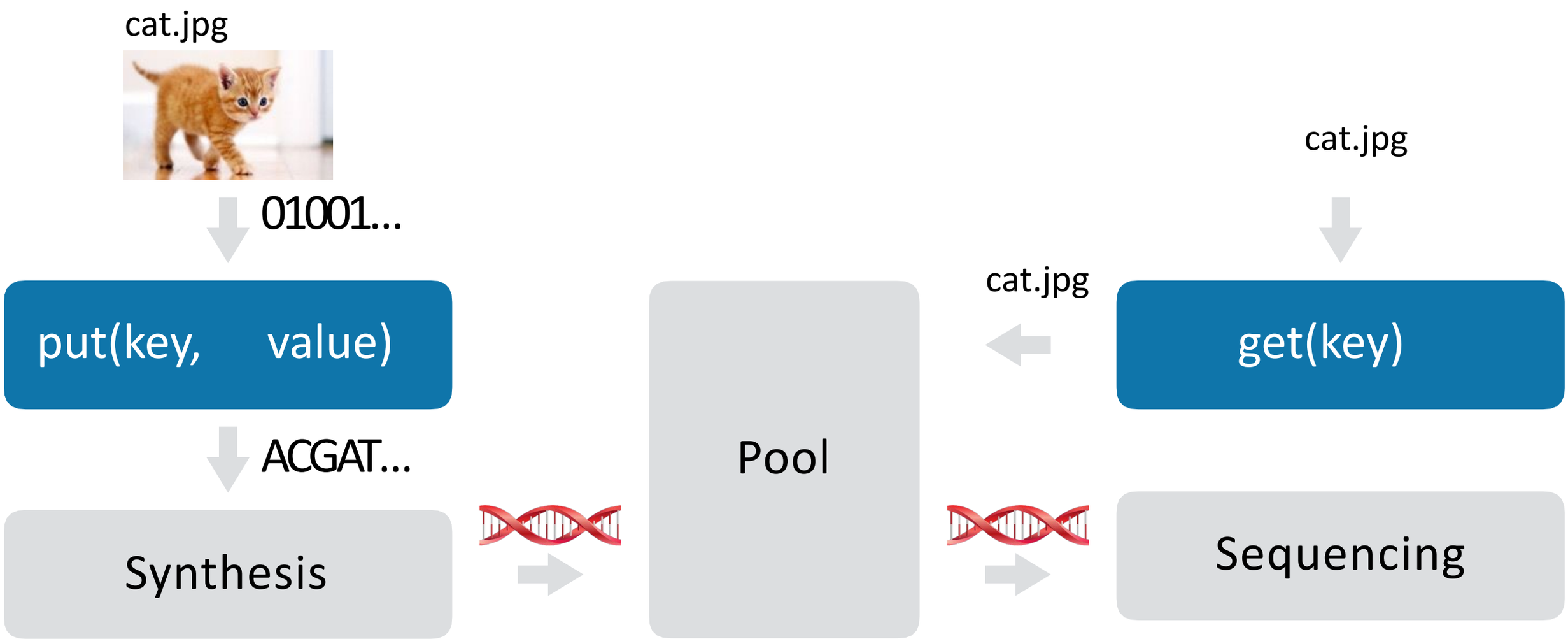
System overview

Archival storage system structured as a key-value store



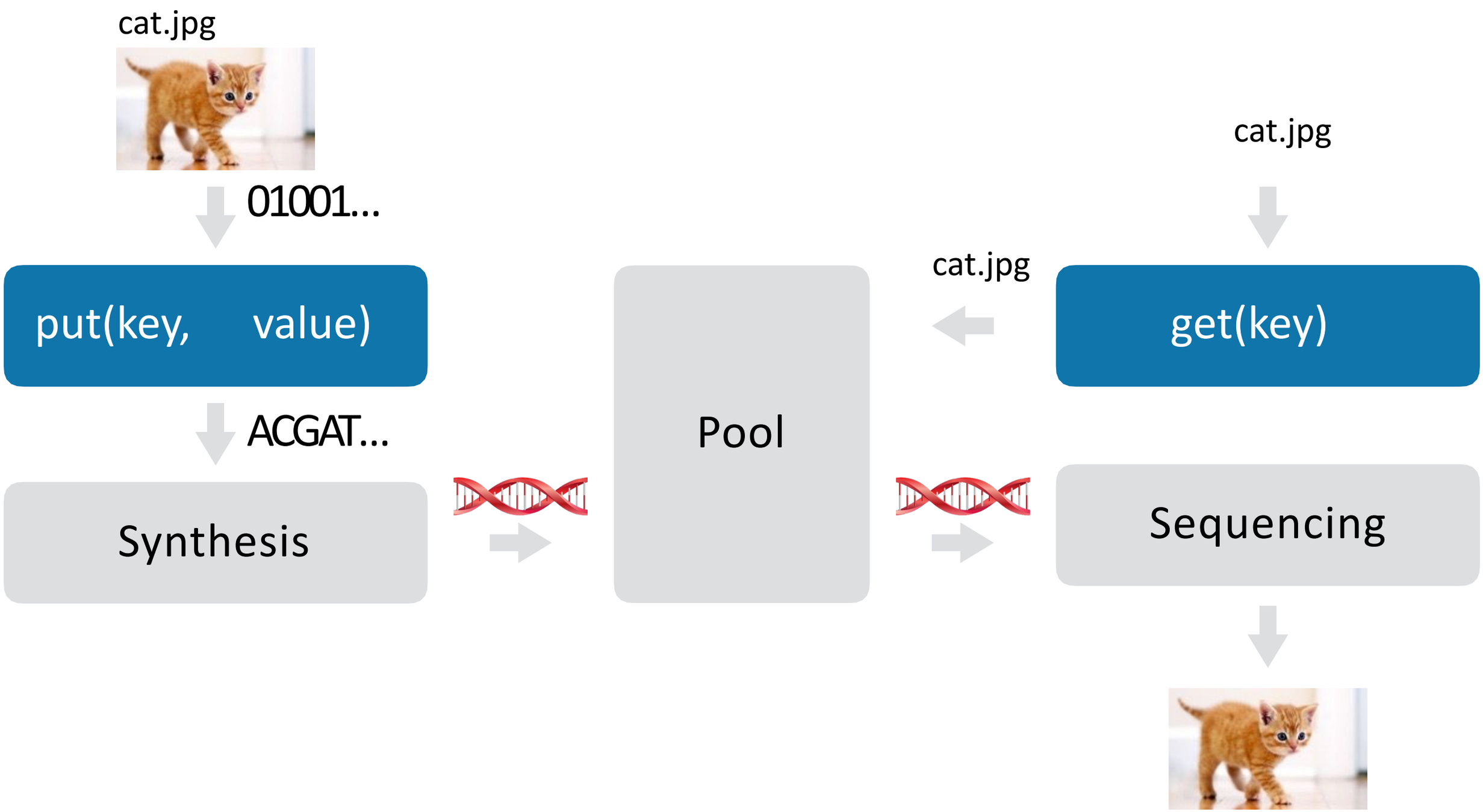
System overview

Archival storage system structured as a key-value store



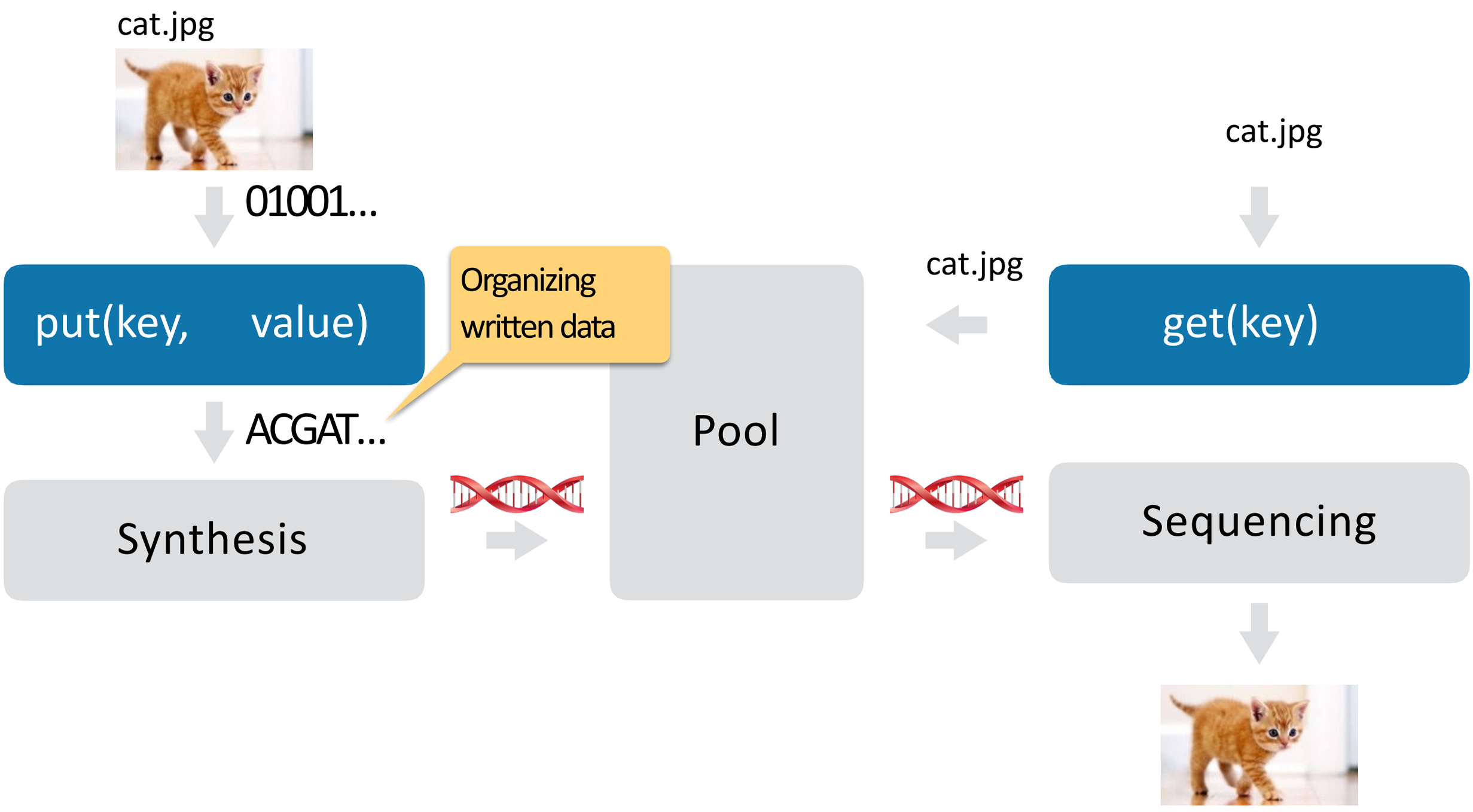
System overview

Archival storage system structured as a key-value store



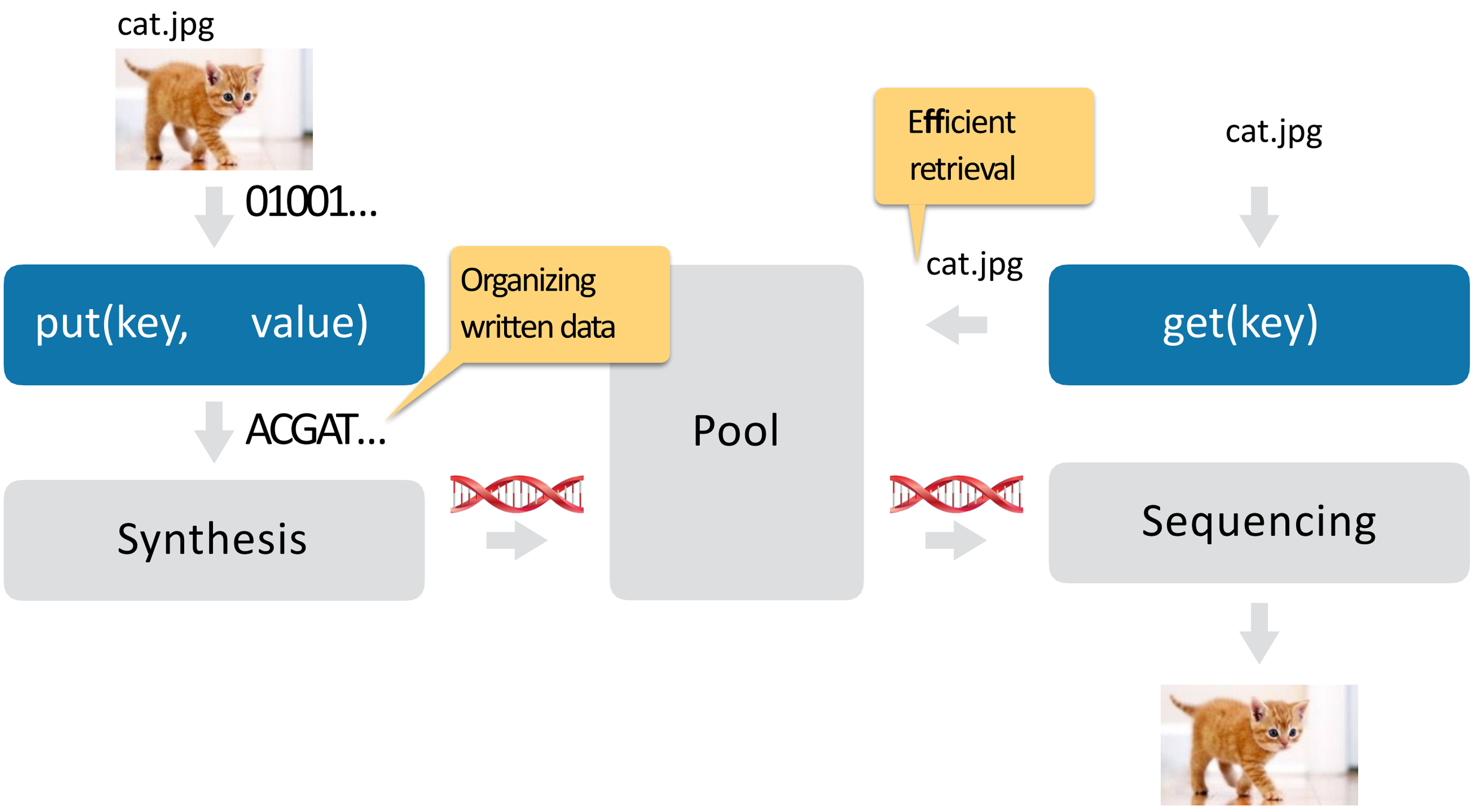
System overview

Archival storage system structured as a key-value store



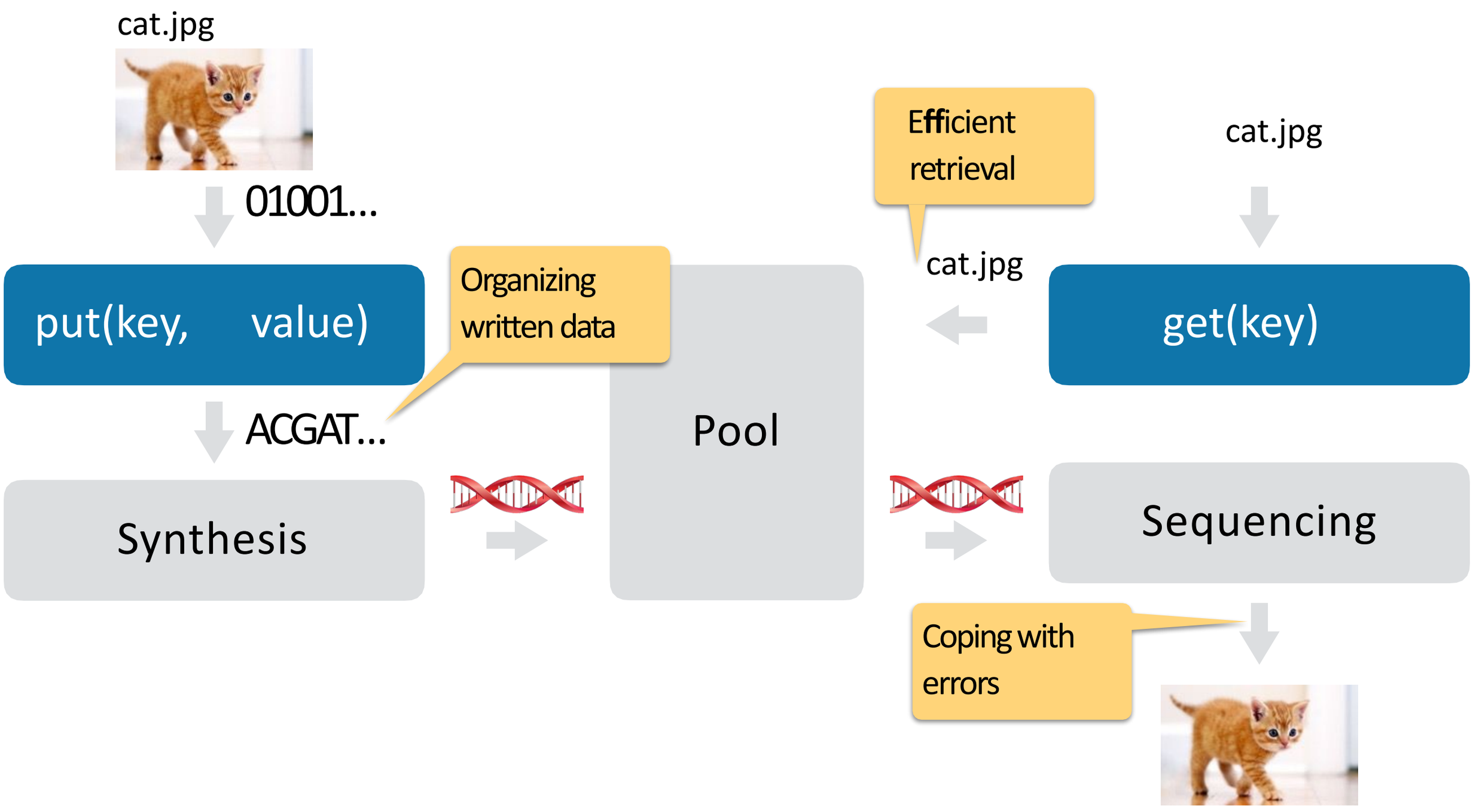
System overview

Archival storage system structured as a key-value store



System overview

Archival storage system structured as a key-value store



Overall, DNA-based storage has potential as a practical solution to the digital archiving problem and may become a cost-effective solution for rarely accessed archives.

We hope you enjoyed learning something about biology, computer science, electrical Engineering and the creative combination of all for a data storage system.

Thanks for reading! 😊