Ultra96 Design

|  |  |
| --- | --- |
| **On CPU** | **On FPGA** |
| Input |  |
|  | CDC |
|  | SHA |
| LZW |  |
| Output |  |

# I/O

**Payload Size**

**Our implementation uses a payload size of 8192 Bytes. We have explored the continuum of payload sizes, as small as 256 Bytes, and as large as the Client would allow.**

**Based on our findings, we believe that the payload size vs compression ratio curve hits an inflection point around 8192 Bytes. Any payload size under 8192 Bytes leads to a compression ratio at or below 1, depending on the input file size. Larger files tend to lead to better compression ratios. Payload sizes over 8192 Bytes start to take too much storage on the CPU and FPGA end, leading to unpredictable behavior.**

**Hence, we decided to stick with this payload size for both our FPGA and CPU implementations.**

|  |  |
| --- | --- |
| *Challenge* | Increasing the payload size reduces memory transfer overhead between calls to the pipeline at the cost of increased storage for larger payloads. |
| *Opportunity* | Increasing the payload size allows LZW more opportunity to compress duplicate occurrences of input chunks and gives CDC more opportunities to identify the best chunk boundaries to maximize duplicates. |
| *Continuum* | Reducing payload size increases the number of CDC, SHA and LZW computations but reduces storage and memory transfer |

**Reading Packets**

**The number of packets our FPGA implementation reads in from the server currently equals the number of CDC compute units we decide to instantiate.**

**This design choice is guided in our belief that**

**Based on our findings, we believe that the payload size vs compression ratio curve hits an inflection point around 8192 Bytes. Any payload size under 8192 Bytes leads to a compression ratio at or below 1, depending on the input file size. Larger files tend to lead to better compression ratios. Payload sizes over 8192 Bytes start to take too much storage on the CPU and FPGA end, leading to unpredictable behavior.**

**Hence, we decided to stick with this payload size for both our FPGA and CPU implementations.**

|  |  |
| --- | --- |
| *Challenge* | Increasing the payload size reduces memory transfer overhead between calls to the pipeline at the cost of increased storage for larger payloads. |
| *Opportunity* | Increasing the payload size allows LZW more opportunity to compress duplicate occurrences of input chunks and gives CDC more opportunities to identify the best chunk boundaries to maximize duplicates. |
| *Continuum* | Reducing payload size increases the number of CDC, SHA and LZW computations but reduces storage and memory transfer |

# FPGA Acceleration

CDC

**Sliding Window Size**

|  |  |
| --- | --- |
| *Challenge* |  |
| *Opportunity* |  |
| *Continuum* |  |

**Storing CDC chunks**

SHA