



**Birzeit University**

**Department of Electrical & Computer Engineering**

**Second Semester - 2023/2024**

**ENCS5337 Chip Design Verification**

**Course Project**

**Design Verification of a Hardware Compression and Decompression Chip**

## Chip Specifications

The chip can do both data compression and decompression. It implements a dictionary based compression algorithm.

## Compression Algorithm

The compression algorithm works as follows:

1. The chip receives the input data to be compressed on the **data\_in** port
2. It compares this input data with the stored data in the chip's internal memory (dictionary memory)
3. If the input data is found, the index of the stored similar data is written on the **compressed\_out** port. This index value is the compressed version of this input data.
4. If the input data is not found, the input data is written at the last empty slot of the dictionary memory, then the index is written on the **compressed\_out** port, and the index is incremented
5. The physical index register width is 32-bit. However, the number of bits used out of these 32 bits depends on the size of the dictionary memory. In this project, the default size of the dictionary memory is 256 locations (each

location is 80-bit). Thus, we need the least significant eight bits of these 32 bits, and therefore the compressed data size is 8 bits.

6. If the internal memory is full, the output response signal indicates that there is an error.

### **Decompression Algorithm**

The decompression algorithm is the reverse of the compression algorithm, and it works as follows:

1. The chip receives the compressed data on the **compressed\_in** port
2. If the value of the received compressed data is less than or equal the current value of the index register, then the corresponding decompressed data exists in the dictionary memory at an index equals the value of the received compressed data. Then, the content of the dictionary memory at that index is written on the **decompressed\_out** port.
3. If the value of the received compressed data is greater than the current value of the index register, then the corresponding decompressed data does not exist in the dictionary memory. Therefore, an error is reported.

Table 1 lists the different input/out ports of this compression/decompression chip

Table 1: Compression/Decompression Chip Input/output Ports

Port	Default Width (#bits)	Direction	Description
clk	1	Input	
reset	1	Input	Clears the dictionary memory and the index register
command	2	input	Specifies the chip operation  00: No operation 01: Compression 10: Decompression 11: Invalid command, report an error
data_in	80	input	Data to be compressed
compressed_in	8	input	Data to be decompressed
compressed_out	8	Output	Output compressed data
decompressed_out	80	Output	Output decompressed data
response	2	Output	Shows the status of the output  00: no valid output 01: valid compressed_out 10: valid deccompressed_out 11: Error

Figure 1 shows a high-level block diagram of the compression/decompression chip

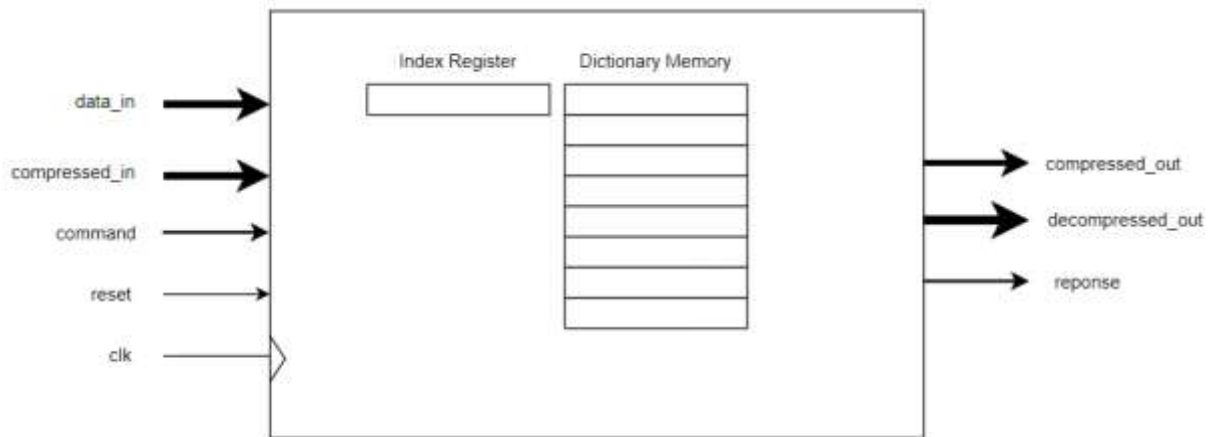


Figure 1: Compression/Decompression Chip Block Diagram

## Project Phases

	Weight	Deadline	Deliverables
Phase1: Reference Model	8%	20/4/2024	<ul style="list-style-type: none"><li>- Report</li><li>- Reference model code</li><li>- Demo and Discussion</li></ul>
Phase2: Verification Plan	7%	5/5/2024	<ul style="list-style-type: none"><li>- Complete verification plan document</li></ul>
Phase3: Complete UVM Verification Environment with Coverage	20%	1/6/2024	<ul style="list-style-type: none"><li>- Detailed report</li><li>- Fully documented verification code</li><li>- Fully working verification environment</li><li>- Simulation results</li><li>- Detailed demonstration and discussion</li></ul>