

LZ77 DECODING CIRCUIT AND WORKING METHOD THEREOF

Field of the Invention

5 The present disclosure discloses an LZ77 decoding circuit and a working method thereof, belonging to the technical field of data decoding.

Background of the Invention

10 LZ77 is a dictionary-based algorithm, which uses encoded information as a dictionary. If a string to be encoded once appeared, a position where the string appeared and a length of the string are configured to replace the string so as to fulfill an aim of compression. The LZ77 algorithm is applied to almost all daily used universal compression tools, for example, ARJ, PKZip, WinZip, LHArc, RAR and GZip, and compression algorithms built in many types of
15 hardware.

 Data is compressed by LZ77 into two forms, namely literal and pair (length, distance). Literal is a character which is not compressed, and pair is a two-dimensional tuple which includes two pieces of information, namely length and distance, and represents a string
20 sequence with an unfixed length. During data decompression, literal does not need special processing, whereas pair requires data recovery through a search window (search-window), so that a pair processing speed of a circuit determines a decoding speed of the entire circuit.

 Chinese patent document CN109716659A discloses a high-performance single-stream
25 LZ77 compression technology. The high-performance single-stream data compression technology uses a computing device which updates an index data structure on the basis of an input data stream. The input data stream is divided into many blocks. Each of the blocks has a predetermined length, for example 136 bytes, and is overlapped with the previous block by a predetermined quantity, for example, 8 bytes. The computing device processes a plurality of

blocks in parallel using index data to generate a plurality of token streams. Tokens include literal tokens and reference tokens, and the reference tokens refer to relatively early matched data in the input data stream. The document can effectively improve the decoding efficiency of the LZ77 algorithm which is a core component of the existing Gzip decompression algorithm.

5 However, this document does not provide corresponding address mapping circuit functional modules and output preprocessing circuit functional modules. Therefore, it is difficult to realize visual control of the pair processing speed of the circuit during data recovery.

Summary of the Invention

10

To solve the above technical problems, the present disclosure discloses an LZ77 decoding circuit.

The present disclosure further discloses a working method of the decoding circuit.

15

The present disclosure respectively pre-processes input data and output data through an improved search window structure, thereby effectively enhancing the LZ77 decoding speed.

A detailed technical solution of the present disclosure is as follows.

20

An LZ77 decoding circuit is provided; in the circuit, data to be decompressed are data which can be recovered to include literal and pair; since LZ77 compression is usually used in combination with other compression algorithms to compress data by a plurality of stages, but the decoding circuit of the present disclosure can be used to work only when the data to be
25 decompressed can be recovered into data in form of literal or pair during LZ77 decoding; as shown in Fig. 10 and Fig. 11, taking GZIP decompression as an example, data to be decompressed are decompressed by a Huffman decoding circuit first and then decompressed by LZ77;

according to a direction of a pair data stream, the LZ77 decoding circuit includes:

an input/output control circuit, an input buffer, an input preprocessing circuit, an address mapping circuit, a RAM input/output control circuit, a RAM, an output preprocessing circuit and an output buffer, wherein the RAM and the output buffer jointly form a search window, and the RAM input/output control circuit searches for data required by pair from the RAM of the search window;

according to a direction of a literal data stream, the LZ77 decoding circuit includes:

an input buffer, an input preprocessing circuit and an output buffer, wherein literal data in the data to be decompressed represents a part which is not compressed by LZ77, and the literal data are directly placed into the output buffer by the input preprocessing circuit;

data in the output buffer updates the RAM, all decompressed data and the literal data which are directly placed in the output buffer are stored in the output buffer first, and meanwhile the output buffer counts the data, and writes corresponding data into the RAM of the search window through the RAM input/output control circuit for storage when a data size reaches a certain threshold value to realize RAM update, as shown in Fig. 1;

the input/output control circuit is configured to control input of data to be decompressed and data output after decompression;

the input buffer is configured to buffer the data input by the input/output control circuit, a lower limit of the size of the input buffer is required to be defined in the input buffer, but the lower limit of the size of the input buffer is not a fixed value but related to $\max(\text{literal_length}, \text{pair_length})$, namely the maximum value in a literal bit length or a pair bit (distance+length) length; furthermore, the input buffer is required to buffer the longest one of literal or pair;

the input preprocessing circuit is configured to pre-process data to be decompressed into literal data and/or pair data; in the art, literal and pair are distinguished with certain “marks” in

a specific compression algorithm; taking the application of LZ77 in the GZIP compression algorithm as an example, literal is represented by 0-255, while length (a first component in pair) is represented by 257-285 plus an extension byte, so literal and length can be distinguished through the data range during decompression, length and distance appear in pair, and the data behind length is distance;

the address mapping circuit is configured to parse pair data; if data required by pair data are entirely or partly located in the RAM (length pieces of data are directly copied from the output buffer when all required data are located in the output buffer); distance in the pair data is mapped as an address (ram_rd_st_addr) which can be identified by the RAM, the count (ram_rd_cnt) of reading data from the RAM is calculated according to length in the pair data, and the position, count and length information of data to be searched for are determined according to the number (out_buf_cnt) of data in the output buffer and the address (ram_cur_addr) currently directed to by the RAM;

the output preprocessing circuit reads back data from the RAM according to the position, count and length information of the data, shears and splices data, and finally forms decompressed data;

the output buffer is configured to buffer the decompressed data;

the RAM input/output control circuit is configured to control writing data into and reading data from the RAM; and

the search window includes the RAM and the output buffer, and the output buffer updates the RAM.

According to the present disclosure, preferably, the search window includes a dual-port RAM and an output buffer; the dual-port RAM in the search window of the minimum

architecture has a size the same as that of a “dictionary” used during compression with LZ77. For example, if a “dictionary” with a size of 32 KBytes is used during compression with LZ77 in the GZIP algorithm, the size of the RAM is required to be the same as that of the “dictionary”; and the size of the output buffer itself is greater than or equal to a data width corresponding to each of addresses of the RAM. The data width corresponding to each of the addresses of the RAM is nBytes. For example, a 32-KByte RAM is in form of 4K*8 Bytes, where n=8, and then the size of the output buffer is required to be greater than or equal to 8 Bytes. During decompression, the output buffer counts the decompressed data, writes the 8-Byte data into the RAM through the RAM input/output control module every time the count value is greater than or equal to 8 Bytes, and outputs the data via the input/output control module at the same time; and then, the data will be discarded by the output buffer by means of displacement, and the count value is correspondingly deducted by 8. This technical characteristic avoids waste in clock resources caused by frequent reading from and writing into the RAM, can read multi-byte data within one clock cycle during pair decompression, and fully utilizes the RAM and an output buffer circuit, thereby increasing the decoding speed.

A working method of the decoding circuit is provided, including

1) reading data to be decompressed and placing the data into the input buffer by the decoding circuit;

2) preprocessing the data to be decompressed into literal data and/or pair data, and transmitting the literal data into the output buffer and the pair data into the address mapping circuit;

3) mapping distance in the pair data as an address (ram_rd_st_addr) which can be identified by the search window, computing the count (ram_rd_cnt) of reading data from the search window according to length in the pair data, and determining position, count and length information of data to be searched for according to the number of data (out_buf_cnt) in the output buffer and the address (ram_rd_st_addr) currently directed to by the RAM;

4) reading back data from the search window according to the position, count and length information of the data to be searched for;

5) performing output preprocessing on the data read back: shearing and splicing data, and
5 finally obtaining the decompressed data; and,

6) repeating steps 1) to 5) until decompression of all data to be decompressed is completed.

According to the present disclosure, preferably, the working method further includes:

10

transmitting the literal data and the decompressed data into the output buffer, counting the data, and after a data size reaches a preset threshold value, writing the corresponding data into the RAM for storage and updating the RAM by the output buffer.

15

During decompression, a single matched string can exist at the following three positions:

in the RAM only; or

in the output buffer only; or

20

in both the RAM and the output buffer at the same time.

According to the present disclosure, preferably, the literal data and the decompressed data are transmitted into the output buffer, and when processing of data to be decompressed once is completed, the data in the output buffer are outputted.

25

According to the present disclosure, preferably, updating the RAM refers to rolling update,

including:

5 writing data from a low address to a high address or from the high address to the low address in turn when data in the output buffer are written into the RAM. Taking writing data into a 4K*8-Byte RAM from the high address to the low address as an example, ram_cur_addr with a bit width of 12 bit represents the address that next data are written; when decompression begins, ram_cur_addr is required to direct to the highest address 4095; every time after a piece of 8-Byte data are written, the address is deducted by 1; and when ram_cur_addr is deducted to 0, ram_cur_addr is deducted by 1 again to go back to 4095 due to the binary counting
10 characteristic. Thus, rolling update of the RAM is realized. Similarly, writing from the low address, namely 0, is also feasible, but ram_cur_addr needs to be added by 1 every time data are written. Such operation can also achieve the effect of rolling update of the RAM.

15 According to the present disclosure, preferably, output preprocessing in step 5) includes:

shearing data to remove redundant data; and

splicing data to re-splice the sheared data and subsequent data in an aligned format.

20 Since a string represented by the pair data is not fixed in length, for example in the GZIP algorithm, length ranges from 3 to 258 Bytes, and the data read back from the search window through the RAM are all data of this address, redundant data need to be removed. As shown in Fig. 9, if only four Bytes, namely c, d, e and f, are matched with the pair data, but all Bytes, including a, b, c, d, e, f, g and h, of the address are read out during operation on the RAM, all
25 data other than c, d, e and f which are effective data need to be removed. The sheared data are spliced to a high address of the output buffer, and as the output buffer includes three Bytes, namely x, y and z, the output buffer includes x, y, z, c, d, e and f after splicing for RAM update and data output. At this time, the decompression operation on single pair data is completed.

According to the present disclosure, preferably, mapping rules of the address mapping circuit in step 2) include:

1) the address mapping circuit determines the size of distance in the pair data;

5

1-1) if distance is smaller than or equal to out_buf_cnt, length pieces of data are directly copied from the output buffer;

1-2) if distance is greater than out_buf_cnt, data are copied from the RAM and the size of
10 length is determined;

1-2-1) if length is smaller than or equal to distance – out_buf_cnt, the data to be searched for exist in the RAM only,

15 starting address (ram_rd_st_addr) = ram_rd_st_addr - [(distance-out_buf_cnt)/number of bytes corresponding to each of addresses of the RAM]_{rounded up to an integer}, count (rd_cnt);

if length is less than or equal to [(distance-out_buf_cnt)/number of bytes corresponding to each of address of the RAM]_{remainder}, the RAM operation is carried out once only, rd_cnt=1;

20

if length is greater than [(distance-out_buf_cnt)/number of bytes corresponding to each of addresses of the RAM]_{remainder}, rd_cnt = [(length - [(distance-out_buf_cnt)/number of bytes corresponding to each of address of the RAM]_{remainder})/number of bytes corresponding to each of address of the RAM]_{rounded up to an integer};

25

reading process: the reading operation on the RAM begins from the starting address, ram_rd_st_addr is reduced by 1 every time reading is carried out, then the reading operation is

implemented on new ram_rd_st_addr until reading for rd_cnt times is completed, and there may be redundancy in data read out from the RAM at the first time and at the last time; and,

5 1-2-2) if length is greater than distance – out_buf_cnt, distance – out_buf_cnt pieces of data are searched from the RAM, and the remaining length -distance -out_buf_cnt pieces of data are obtained from the output buffer.

10 The above are address mapping rules. In addition, data read out from the RAM are further required to be sheared and spliced by the output preprocessing circuit before finally entering the output buffer.

The technical advantages of the present disclosure are as follows:

15 1. The present disclosure can be widely applied to the LZ77 decoding portion in various decompression algorithms; the present disclosure needs to design corresponding input preprocessing circuit only according to the “marks” of literal and pair in different algorithms, and selects RAMs of different sizes according to the size of the “dictionary” used during compression. This operation can be universally applied to other circuit parts, thereby facilitating duplication and improving reusability of the circuit.

20

2. Component structure and update method of the search window of the present disclosure: the RAM and the output buffer are configured to jointly form the search window, and the operation of writing data into the RAM is carried out once only when the size of the data in the output buffer reaches a certain threshold value. Such operation method avoids waste in clock resources caused by frequent reading from and writing into the RAM, and can read multi-byte data within one clock cycle during decompression of the pair data, thereby fully utilizing the RAM and the output buffer, and increasing the decoding speed.

25

3. The present disclosure can quickly obtain decompressed data by providing the mapping

rules of the address mapping circuit matched with the search window and the output preprocessing circuit.

Brief Description of the Drawings

5

Fig. 1 is a structure diagram of an LZ77 decoding circuit according to the present disclosure;

Fig. 2 is a flowchart of data decompression according to the present disclosure;

10

Fig. 3 is a diagram of an input preprocessing circuit according to the present disclosure;

Fig. 4 is a schematic diagram of single writing into a RAM according to the present disclosure;

15

Fig. 5 is a schematic diagram of rolling update of the RAM according to the present disclosure;

Fig. 6 is a schematic diagram of a search window according to the present disclosure;

20

Fig. 7-1 is a schematic diagram of data which are matched with pair and only exist in the RAM according to the present disclosure;

Fig. 7-2 is a schematic diagram of data which are matched with pair and only exist in an output buffer according to the present disclosure;

25

Fig. 7-3 is a schematic diagram of data which are matched with pair and exist in both the RAM and the output buffer at the same time according to the present disclosure;

5 Fig. 8 is a schematic diagram of an address mapping circuit according to the present disclosure;

Fig. 9 is a schematic diagram of shearing and splicing of data read back according to the present disclosure;

10 Fig. 10 is a schematic diagram of an application scenario of an LZ77 decoding circuit in GZIP decoding according to the present disclosure; and

Fig. 11 is a flowchart of the application scenario of the LZ77 decoding circuit in GZIP decoding according to the present disclosure.

15

Detailed Description of the Embodiments

The present disclosure will be described in detail below with reference to embodiments and accompany drawings, but is not limited thereto.

20

Embodiment 1

As shown in Figs. 1, 2 and 3,

an LZ77 decoding circuit is provided; in the circuit, data to be decompressed are data
25 which may be recovered to include literal and pair;

according to a direction of a pair data stream, the LZ77 decoding circuit includes:

an input/output control circuit, an input buffer, an input preprocessing circuit, an address mapping circuit, a RAM input/output control circuit, a RAM, an output preprocessing circuit and an output buffer, the RAM and the output buffer jointly form a search window; and the RAM input/output control circuit searches for data required by pair from the RAM of the search window;

according to a direction of a literal data stream, the LZ77 decoding circuit includes:

an input buffer, an input preprocessing circuit and an output buffer; and data in the output buffer update the RAM;

the input/output control circuit is configured to control input of data to be decompressed and data output after decompression;

the input buffer is configured to buffer data input by the input/output control circuit.

the input preprocessing circuit is configured to pre-process data to be decompressed into literal data and/or pair data;

the address mapping circuit is configured to parse the pair data, maps distance in the pair data as an address (ram_rd_st_addr) which may be identified by the search window, computes a count (ram_rd_cnt) of reading data from the RAM according to length in the pair data, and determines position, count and length information of data to be searched for according to the number of data (out_buf_cnt) in the output buffer and the address (ram_rd_st_addr) currently directed to by the RAM;

the output preprocessing circuit reads back data from the RAM according to the position, count and length information of the data, shears and splices data, and finally forms decompressed data;

5 the output buffer is configured to buffer the decompressed data;

the RAM input/output control circuit is configured to control writing data into and reading data from the RAM; and

10 the search window includes the RAM and the output buffer, and the output buffer updates the RAM, as shown in Fig. 6.

Embodiment 2

15 As shown in Fig. 5, according to the LZ77 decoding circuit as described in Embodiment 1, the search window includes a dual-port RAM and an output buffer; the dual-port RAM in the search window of the minimum architecture has a size the same as that of a “dictionary” used during compression with LZ77, for example, if a “dictionary” with a size of 32 KBytes is used during compression with LZ77 in the GZIP algorithm, the size of the RAM is required to be the same as that of the “dictionary”; and the size of the output buffer itself is greater than or equal to a data width corresponding to each of addresses of the RAM. The data width corresponding to each of the addresses of the RAM is nByte. For example, a 32-KByte RAM is in form of 4K*8 Bytes, where n=8, and then the size of the output buffer is required to be greater than or equal to 8 Bytes. During decompression, the output buffer counts the decompressed data, writes the 8-Byte data into the RAM through the RAM input/output control module every time the count value is greater than or equal to 8 Bytes, and outputs the data via the input/output control module at the same time; and then, the data will be discarded by the output buffer by means of displacement, and the count value is correspondingly deducted by 8. This technical characteristic avoids waste in clock resources caused by frequent reading from and writing into the RAM, and can read multi-Byte data within one clock cycle during pair decompression,

20
25
30

thereby fully utilizing the RAM and an output buffer circuit, and then increasing the decoding speed.

Embodiment 3

5

The working method of the decoding circuit according to Embodiments 1 and 2 includes:

1) data to be decompressed are read and placed into the input buffer by the decoding circuit;

10 2) the data to be decompressed are preprocessed into literal data and/or pair data, the literal data are transmitted into the output buffer and the pair data are transmitted into the address mapping circuit;

15 3) distance in the pair data is mapped as an address (ram_rd_st_addr) which may be identified by the search window, the count (ram_rd_cnt) of reading data from the search window is computed according to length in the pair data, and position, count and length information of data to be searched for are determined according to the number of data (out_buf_cnt) in the output buffer and the address (ram_rd_st_addr) currently directed to by the RAM, as shown in Fig. 8;

20

4) data are read back from the search window according to the position, count and length information of the data to be searched for;

25 5) output preprocessing is performed on the data read back: data are sheared and spliced, and finally the decompressed data are obtained; and

6) steps 1) to 5) are repeated until decompression of all data to be decompressed is

completed.

Embodiment 4

5 The working method of the decoding circuit according to Embodiment 3 further includes:

the literal data and the decompressed data are transmitted into the output buffer, the data are counted, and after the data size reaches a preset threshold value, the corresponding data are written into the RAM for storage and the RAM is updated by the output buffer.

10

During decompression, a single matched string may exist at the following three positions:

in the RAM only; or

15 in the output buffer only; or

in both the RAM and the output buffer at the same time, as shown in Figs. 7-1, 7-2 and 7-3.

20 The literal data and the decompressed data are transmitted into the output buffer, and when processing of data to be decompressed once is completed, the data in the output buffer are outputted.

Updating the RAM refers to rolling update, as shown in Figs. 4 and 5, including:

25 data are written from a low address to a high address or from the high address to the low address in turn when data in the output buffer are written into the RAM. Taking writing data

into a 4K*8-Byte RAM from the high address to the low address as an example, ram_cur_addr with a bit width of 12 bit represents the address that next data are written; when decompression begins, ram_cur_addr is required to direct to the highest address 4095; every time after a piece of 8-Byte data is written, the address is deducted by 1; and when ram_cur_addr is deducted to 0, ram_cur_addr is deducted by 1 again to go back to 4095 due to the binary counting characteristic. Thus, rolling update of the RAM is realized. Similarly, writing from the low address, namely 0, is also feasible, but ram_cur_addr needs to be added by 1 every time data are written. Such operation can also achieve the effect of rolling update of the RAM.

Output preprocessing in step 5) includes:

data are sheared to remove redundant data; and

data are spliced to re-splice the sheared data and subsequent data in an aligned format.

Since a string represented by the pair data is not fixed in length, for example in the GZIP algorithm, length ranges from 3 to 258 Bytes, and the data read back from the search window through the RAM are all data of this address, redundant data need to be removed. As shown in Fig. 9, if only four Bytes, namely c, d, e and f, are matched with the pair data, but all Bytes, including a, b, c, d, e, f, g and h, of the address are read out during operation on the RAM, all data other than c, d, e and f which are effective data need to be removed. The sheared data are spliced to a high address of the output buffer, and as the output buffer includes three Bytes, namely x, y and z, the output buffer includes x, y, z, c, d, e and f after splicing for RAM update and data output. At this time, the decompression operation on single pair data is completed.

Embodiment 5

According to Embodiment 3, mapping rules of the address mapping circuit in step 2) include:

1) the address mapping circuit determines the size of distance in the pair data;

1-1) if distance is smaller than or equal to out_buf_cnt, length pieces of data are directly copied from the output buffer;

5

1-2) if distance is greater than out_buf_cnt, data are copied from the RAM and the size of length is determined;

1-2-1) if length is smaller than or equal to distance – out_buf_cnt, the data to be searched
10 for exist in the RAM only,

starting address (ram_rd_st_addr) = ram_rd_st_addr - [(distance-out_buf_cnt)/number of bytes corresponding to each of addresses of the RAM]_{rounded up to an integer}, count (rd_cnt);

15 if length is less than or equal to [(distance-out_buf_cnt)]/number of bytes corresponding to each of address of the RAM]_{remainder}, RAM operation is carried out once only, rd_cnt=1;

if length is greater than [(distance-out_buf_cnt)]/number of bytes corresponding to each of addresses of the RAM]_{remainder}, rd_cnt = [(length - [(distance-out_buf_cnt)/number of bytes corresponding to each of address of the RAM]_{remainder})]/number of bytes corresponding to each
20 of address of the RAM]_{rounded up to an integer};

reading process: the reading operation on the RAM begins from the starting address, ram_rd_st_addr is reduced by 1 every time reading is carried out, then the reading operation is implemented on new ram_rd_st_addr until reading for rd_cnt times is completed, and there may
25 have redundancy in data read out from the RAM at the first time and at the last time; and

1-2-2) if length is greater than distance – out_buf_cnt, distance – out_buf_cnt pieces of data are searched from the RAM, and the remaining length -distance -out_buf_cnt pieces of

data are obtained from the output buffer.

The above are address mapping rules. In addition, data read out from the RAM are further required to be sheared and spliced by the output preprocessing circuit before finally entering the output buffer.

The present disclosure provides the address mapping circuit and the mapping rules, such that length and distance in the pair data may be adapted to the search window structure of the present disclosure, and meanwhile the output preprocessing circuit shears and splices redundant information brought by such search window structure to quickly obtain the string matched with pair.

If the data width of the RAM is 1 Byte, then at least length pieces of clock cycles are required to process the length pieces of pair data. This is because even if a circuit of a pipeline architecture is used, the reading operation consumes at least one clock cycle each time, thus limiting the upper limit of the decoding speed to be less than 1 Byte/cycle. The present disclosure properly increases the data width corresponding to each of addresses of the RAM to enhance the upper limit of the decoding speed, and provides a corresponding RAM update method to increase the decoding speed. For example, if the size of the RAM is 32 KBytes (4K*8 Bytes), since the data width of each of addresses of the RAM is 8 Bytes, and the RAM is updated every time one Byte is decompressed, then a reading operation is required before writing of this address. For example, if 8 pieces of literal are decompressed continuously, for example, a sequence “abcdefgh”, updating the RAM proceeds in the following sequence: write into a → read out a → write in ab → read out ab → write in abc → read out abc → write in abcd → read out abcd → write in abcde → read out abcdefg → write into abcdefgh. A total of 15 clock cycles are required to complete such operation, so the time cost is high.

The present disclosure provides a “dual-port RAM + output buffer” structure. If the decompressed data are placed in the output buffer first and counted (out_buf_cnt), and the size of the output buffer is 8 Bytes, then the size of the buffered data needs to reach 8 Bytes before

the operation of RAM update, as shown in Fig. 5. The state of the output buffer changes in the following sequence: $a \rightarrow ab \rightarrow abc \rightarrow abcd \rightarrow abcde \rightarrow abcdef \rightarrow abcdefg \rightarrow abcdefgh \rightarrow$ write in abcdefgh, as shown in Fig. 4. Such operation costs only 9 clock cycles, thereby greatly improving the decoding efficiency.

5

At the same time, the decompressed data are outputted to empty the output buffer. By the above description, it can be known that the search window includes two parts, one part representing data which have been written into the RAM, and the other representing data which are not written into the RAM; and the data in the output buffer are the data which are not written into the RAM. This is specified by a set threshold value.

10

Claims

1. An LZ77 decoding circuit, characterized in that data to be decompressed are capable of being recovered into data comprising literal and pair,

5 wherein according to a direction of a pair data stream, the LZ77 decoding circuit comprises:

an input/output control circuit, an input buffer, an input preprocessing circuit, an address mapping circuit, a RAM input/output control circuit, a RAM, an output preprocessing circuit and an output buffer, wherein the RAM and the output buffer jointly form a search window; and the RAM input/output control circuit searches for data required by pair from the RAM of
10 the search window;

according to a direction of a literal data stream, the LZ77 decoding circuit comprises:

an input buffer, an input preprocessing circuit and an output buffer;

data in the output buffer update the RAM;

the input/output control circuit is configured to control input of data to be decompressed
15 and data output after decompression;

the input buffer is configured to buffer data input by the input/output control circuit;

the input preprocessing circuit is configured to pre-process data to be decompressed into literal data and/or pair data;

the address mapping circuit is configured to parse the pair data; maps distance in the pair
20 data as an address (ram_rd_st_addr) which is capable of being identified by the search window, computes a count (ram_rd_cnt) of reading data from the RAM according to length in the pair data, and determines position, the count and length information of data to be searched for according to the number of data (out_buf_cnt) in the output buffer and the address (ram_rd_st_addr) currently directed to by the RAM;

25 the output preprocessing circuit reads back data from the RAM according to the position, count and length information of the data, shears and splices data, and finally forms decompressed data;

the output buffer is configured to buffer the decompressed data;

the RAM input/output control circuit is configured to control writing data into and reading data from the RAM; and,

the search window comprises the RAM and the output buffer, wherein the output buffer updates the RAM.

5

2. The LZ77 decoding circuit according to claim 1, characterized in that the search window comprises a dual-port RAM and an output buffer; the dual-port RAM in the search window of the minimum architecture has a size the same as that of a “dictionary” used during compression with LZ77; and a size of the output buffer itself is greater than or equal to a data width
10 corresponding to each of addresses of the RAM.

3. A working method of the decoding circuit according to claim 1 or 2, characterized by comprising:

1) reading data to be decompressed and placing the data into the input buffer by the
15 decoding circuit;

2) preprocessing the data to be decompressed into literal data and/or pair data, and transmitting the literal data into the output buffer and the pair data into the address mapping circuit;

3) mapping distance in the pair data as an address (ram_rd_st_addr) which is capable of
20 being identified by the search window, computing a count (ram_rd_cnt) of reading data from the search window according to length in the pair data, and determining position, count and length information of data to be searched for according to the number of data (out_buf_cnt) in the output buffer and the address (ram_rd_st_addr) currently directed to by the RAM;

4) reading back data from the search window according to the position, count and length
25 information of the data to be searched for;

5) performing output preprocessing on the data read back: shearing and splicing data, and finally obtaining the decompressed data; and,

6) repeating steps 1) to 5) until decompression of all data to be decompressed is finished.

4. The working method of the decoding circuit according to claim 3, characterized in that the working method further comprises:

transmitting the literal data and the decompressed data into the output buffer, and counting the data, and after a data size reaches a preset threshold value, writing the corresponding data
5 into the RAM for storage and updating the RAM by the output buffer,

wherein during decompression, a single matched string is capable of existing at the following three positions:

in the RAM only; or,

in the output buffer only; or,

10 in both the RAM and the output buffer at the same time.

5. The working method of the decoding circuit according to claim 3, characterized in that,

the literal data and the decompressed data are transmitted into the output buffer, and when processing of data to be decompressed once is finished, the data in the output buffer are
15 outputted.

6. The working method of the decoding circuit according to claim 4 or 5, characterized in that updating the RAM refers to rolling update, comprises:

writing data from a low address to a high address or from the high address to the low
20 address in turn when data in the output buffer are written into the RAM.

7. The working method of the decoding circuit according to claim 3, characterized in that output preprocessing in step 5) comprises:

shearing data to remove redundant data; and,

25 splicing data to re-splice the sheared data and subsequent data in an aligned format.

8. The working method of the decoding circuit according to claim 3, characterized in that

mapping rules of the address mapping circuit in step 2) comprise:

1) determining, by the address mapping circuit, the size of distance in the pair data;

1-1) if distance is smaller than or equal to out_buf_cnt, directly copying length pieces of data from the output buffer;

5 1-2) if distance is greater than out_buf_cnt, copying data from the RAM and determining the size of length;

1-2-1) if length is smaller than or equal to distance – out_buf_cnt, determining that the data to be searched for exists in the RAM only,

10 starting address (ram_rd_st_addr) = ram_rd_st_addr - [(distance-out_buf_cnt)/number of bytes corresponding to each of addresses of the RAM]_{rounded up to an integer}, count (rd_cnt);

if length is less than or equal to [(distance-out_buf_cnt)]/number of bytes corresponding to each of address of the RAM]_{remainder}, carrying out RAM operation once only, rd_cnt=1;

15 if length is greater than [(distance-out_buf_cnt)]/number of bytes corresponding to each of addresses of the RAM]_{remainder}, rd_cnt = [(length - [(distance-out_buf_cnt)/number of bytes corresponding to each of address of the RAM]_{remainder}]/number of bytes corresponding to each of address of the RAM]_{rounded up to an integer};

reading process: beginning the reading operation on the RAM from the starting address, reducing ram_rd_st_addr by 1 every time reading is carried out, then carrying out reading operation on new ram_rd_st_addr until reading for rd_cnt times is completed; and

20 1-2-2) if length is greater than distance – out_buf_cnt, searching distance – out_buf_cnt pieces of data from the RAM, and obtaining the remaining length -distance -out_buf_cnt pieces of data from the output buffer.