

NC16FC FPGA IP cores

Readme / Release notes

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice.

This software or any other copies thereof may not be provided or otherwise made available to any other person. No title to and ownership of the software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by the holder of the copyright.

If the Software contains encrypted files you are not allowed to de-crypt or obtain any encrypted information of the delivered files. Any publication or making the source code available to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and publication and will be held liable for any damages caused by violations of the above rules.

Files

nc16fc__readme.txt	this file
nc16fc_modules.html	html ip core description files
nc16fc_modules.txt	text ip core description files
nc16fc_compress.vhd	vhdl generic interface wrapper for compressor
nc16fc_decompress.vhd	vhdl generic interface wrapper for compressor
nc16fc_compress128.sv	compressor with blocksize 128 samples (v1)
nc16fc_decompress128.sv	compressor with blocksize 128 samples (v1)
nc16fc_compress64.sv	compressor with blocksize 64 samples
nc16fc_decompress64.sv	compressor with blocksize 64 samples
nc16fc_compress_pt.sv	passthru with non blocking timing
nc16fc_decompress_pt.sv	passthru with non blocking timing
nc16fc_compress64_pt.sv	passthru with blocksize 64 samples timing
nc16fc_decompress64_pt.sv	passthru with blocksize 64 samples timing

Installation

Xilinx block design does not allow direct connection of system verilog!
Therefore interface wrappers are used to interface the system verilog ip cores.

All interface wrappers are not encrypted and can be edited to match your requirements.
System Verilog (de)compressors are encrypted and can not be viewed or edited.

nc16fc_compress.vhd and nc16fc_decompress.vhd are vhdl generic interface wrapper for the compressor and decompressor. You may use this files to easily upgrade or change the compressor/decompressor. But you are free to use the other vhdl wrappers.

The default (de)compressor is currently set to nc16fc_(de)compress64_v2. nc16fc_(de)compress64_v2 are much smaller and have better overall compression ratio compared to the older (de)compress128_v1. The disadvantage is the slightly higher overhead for incompressible data (5%).

The (de)compress128_v1 ip cores are high performance cores with low overhead of 2% for incompressible data. Other cores with smaller block size and less logic usage will follow. You can use the generic wrappers to exchange the names of the final compressors in the architecture arch_imp section of the generic wrappers file.

All (de)compressors have the same interface signals. Thus the exchange of (de)compressors should be very easy.

To install add all files *.vhd and *.sv to your project and add the required modules in the VIVADO design suite to your block design. Alternatively an ip core can be created from the .vhd and .sv files.

change the name nc16fc_xxx to the name of compressor you want to use in the vhdl section shown below:

```
architecture arch_imp of nc16fc_compress is

COMPONENT nc16fc_xxx is
    PORT
    (
        ...
    );
END COMPONENT;
...
m0 : nc16fc_xxx
    port map
    (
        ...
    );

end arch_imp;
```

Please refer to the Vivado/Labview documentation for more information.

Interface signals are described in the core documentation ncf16.html (one file for all cores)

The system verilog files *.sv are encrypted and can not be viewed or modified

Special Cores

Version 2 now contains special cores for testing

nc16fc_compress64_pt.sv	system verilog passthru with blocksize 64 samples
nc16fc_decompress64_pt.sv	system verilog passthru with blocksize 64 samples

will passthru all data with maximum overhead.
The timing is the same as for nc16fc_(de)compress64.

The cores are subject to change and are included for test purposes only.

License Limit

You need a valid license to use the core for regular data processing. Demo Licenses are generally granted for evaluation and not to compress or decompress normal production data.

If you got a demo license the core is limited to around 420 GSamples for compression.

After this amount of input the core stops compression and transfers packed bytes only. You may then expect compression ratios worse than 0.8.

The limit allow to process data for around 15 min (UHD/30p).

Demo packs are usually marked with demo at the end of the name.

AXIS-Stream compression IP core

Version

Version: 2.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice.

This software or any other copies thereof may not be provided or otherwise made available to any other person. No title to and ownership of the software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by the holder of the copyright.

If the Software contains encrypted files you are not allowed to de-crypt or obtain any encrypted information of the delivered files. Any publication or making the source code available to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and publication and will be held liable for any damages caused by violations of the above rules.

Module

```
module nc16fc_compress128_v1 #  
(  
  
    // Users to add parameters here  
  
    // User parameters ends  
  
    // Do not modify the parameters beyond this line  
  
    // Parameters of Axi Master Bus Interface M00_AXIS  
  
    // Parameters of Axi Slave Bus Interface S00_AXIS  
  
)  
(  
    // Users to add ports here  
  
    input wire      cclk,  
    output wire     status,  
  
    // User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

Compresses the stream with two 16bit words at each sample into a stream which can be decompressed by nc16fc_decompress128_v1

The module needs two AXIS streams which do not honor TKEEP/TSTRB signals. All Input and outputs are 32bit wide.

Input is received from axis stream interface s00_axis_*
Output is send to axis stream interface m00_axis_*

The core needs 2 resets and clocks for the AXIS in/out buses. The clocks must have the same speed. Different clock frequencies are reserved for future use. Resets should be used also the same for the two buses.

Also required for making the core more stable against timing issues is to provide a compute clock (cclk) which must have exactly half of the AXIS clock frequency.

The clock can be connected to the AXIS clocks too but then the total core speed is reduced to 50% (around 80Mhz on Zynq 7K)

The timing of the core is:

- 1.) always 64 samples (128 16bit samples) a read in
on input AXIS/TREADY is hold high during that time
- 2.) after 64 clock cycle TREADY is deasserted for one cycle
- 3.) output starts always after 128 + x cycles
(x is a small number depending on the implementation)
- 4.) the last compressed samples if not aligned to 64 read words
is send to the output if the AXIS/TLAST flag was received on the
input AXIS.
- 5.) if a AXIS/TLAST was received on the input stream the last compressed
word is send with the TLAST signal set on the output AXIS

(this allows to send compressed messages)

The compression core compresses effectively similar 8-16bit samples which are in the stream not too far from each other. This allows to compress video data as well as other data. The core finds out which bits are unused. Thus it can be used also for a packer. E.g. two 8 bit samples presented a input on bit 0-7 and bit 16-23 of the input word are compressed to the output to nearly 50%.

NOTE: all samples (1-16bit half words) must be presented to bit 0-15 and 16-31. E.g. two 12bit samples should be applied to bit 0-11 (first sample) and bit 12-27 (second sample). Otherwise the compressor fail to compress the data.

The status bit is currently unused and is connected to 0. The core can not produce any errors, because all input data can be compressed or packed. The signal is reserved for future use.

Constraints

The core does not need any special constraints.

If you are using it for less than 16bit wide samples it is better to wire the unused bits to GND. This ensures that the unused logic is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper the name of the module is then

```
nc16fc_compress128
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements. You may also include Xilinx specific properties.

AXIS-Stream compression IP core

Version

Version: 2.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used
and copied only in accordance with the terms of such
license and with the inclusion of the above copyright
notice.

This software or any other copies thereof may not be
provided or otherwise made available to any other person.
No title to and ownership of the software is hereby
transferred.

The information in this software is subject to change
without notice and should not be construed as a commitment
by the holder of the copyright.

If the Software contains encrypted files you are not allowed to
de-crypt or obtain any encrypted information of the delivered
files. Any publication or making the source code available
to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and
publication and will be held liable for any damages caused by
violations of the above rules.

Module

```
module nc16fc_compress64_v2 #  
(  
  
    // Users to add parameters here  
  
    // User parameters ends  
  
    // Do not modify the parameters beyond this line  
  
    // Parameters of Axi Master Bus Interface M00_AXIS  
  
    // Parameters of Axi Slave Bus Interface S00_AXIS  
  
)  
(  
    // Users to add ports here  
  
    input wire      cclk,  
    output wire     status,  
  
    // User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

Compresses the stream with two 16bit words at each sample into a stream which can be decompressed by nc16fc_decompress64_v2

The module needs two AXIS streams which do not honor TKEEP/TSTRB signals. All Input and outputs are 32bit wide.

Input is received from axis stream interface s00_axis_*
Output is send to axis stream interface m00_axis_*

The core needs 2 resets and clocks for the AXIS in/out buses. The clocks must have the same speed. Different clock frequencies are reserved for future use. Resets should be the same for the two buses. Resets must be synchronous to the clocks supplied.

Also required for making the core more stable against timing issues is to provide a compute clock (cclk) which must have exactly half of the AXIS clock frequency.

The clock can be connected to the AXIS clocks too but then the total core speed is reduced to 50% (around 80Mhz on Zynq 7K)

The timing of the core is:

- 1.) always 64 samples (64 16bit samples 32 clocks cycles) a read in on input, AXIS/TREADY is hold high during that time
- 2.) after 32 clock cycles TREADY is deasserted for one cycle
- 3.) output starts always after 64 + x cycles
(x is a small number depending on the implementation)
- 4.) the last compressed samples if not aligned to 64 read words is send to the output if the AXIS/TLAST flag was received on the input AXIS.
- 5.) if a AXIS/TLAST was received on the input stream the last compressed word is send with the TLAST signal set on the output AXIS

(this allows to send compressed messages)

The compression core compresses effectively similar 8-16bit samples which are in the stream not too far from each other. This allows to compress video data as well as other data. The core finds out which bits are unused. Thus it can be used also for a packer. E.g. two 8 bit samples presented a input on bit 0-7 and bit 16-23 of the input word are compressed to the output to nearly 50%.

NOTE: all samples (1-16bit half words) must be presented to bit 0-15 and 16-31. E.g. two 12bit samples should be applied to bit 0-11 (first sample) and bit 12-27 (second sample). Otherwise the compressor fail to compress the data.

The status bit is currently unused and is connected to 0. The core can not produce any errors, because all input data can be compressed or packed. The signal is reserved for future use.

Constraints

The core does not need any special constraints.

If you are using it for less than 16bit wide samples it is better to wire the unused bits to GND. This ensures that the unused logic is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper the name of the module is then

```
nc16fc_compress
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements. You may also include Xilinx specific properties.

AXIS-Stream compression IP core

Module

```
module nc16fc_unpack128_bayer12bit #
(

// Users to add parameters here

// User parameters ends

// Do not modify the parameters beyond this line

// Parameters of Axi Master Bus Interface M00_AXIS

// Parameters of Axi Slave Bus Interface S00_AXIS

)
(
// Users to add ports here

input wire      cclk,
output wire     status,

// User ports ends

// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

mpip csi2 bayer12bit unpacker
reads 32bit mipi csi-2 data and outputs 2 x 12bit unpacked data
for compressors

AXIS-Stream compression IP core

Module

```
module nc16fc_compress64_bayer12bit #
(

// Users to add parameters here

// User parameters ends

// Do not modify the parameters beyond this line

// Parameters of Axi Master Bus Interface M00_AXIS

// Parameters of Axi Slave Bus Interface S00_AXIS

)
(
// Users to add ports here

input wire      cclk,
output wire     status,

// User ports ends
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS
input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS
input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

bayer 32bit packed data compressor

this module takes 32bit bayer 12bit packed data read by AXIS s00
unpacks the data and send them to nc16fc_compress64_v2
the output of the compressor is then send via to AXIS M00

The timing of the core blocks after 48 fetched words for 16 clock cycles.
Thus the clock must be 33% higher then needed. for the data transmisson
of raw bayer12bit data.

AXIS-Stream compression IP core

Version

Version: 2.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used
and copied only in accordance with the terms of such
license and with the inclusion of the above copyright
notice.

This software or any other copies thereof may not be
provided or otherwise made available to any other person.
No title to and ownership of the software is hereby
transferred.

The information in this software is subject to change
without notice and should not be construed as a commitment
by the holder of the copyright.

If the Software contains encrypted files you are not allowed to
de-crypt or obtain any encrypted information of the delivered
files. Any publication or making the source code available
to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and
publication and will be held liable for any damages caused by
violations of the above rules.

Module

```
module nc16fc_compress64_pt_v2 #  
(  
  
    // Users to add parameters here  
  
    // User parameters ends  
  
    // Do not modify the parameters beyond this line  
  
    // Parameters of Axi Master Bus Interface M00_AXIS  
  
    // Parameters of Axi Slave Bus Interface S00_AXIS  
  
)  
(  
    // Users to add ports here  
  
    input wire      cclk,  
    output wire     status,  
  
    // User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

A test core which pass through all data without compression.
all timing is the the same as for (de)compress64.

For test only!

Constraints

The core does not need any special constraints.

If you are using it for less than 16bit wide samples it is better
to wire the unused bits to GND. This ensures that the unused logic
is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper
the name of the module is then

```
nc16fc_compress
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements.
You may also include Xilinx specific properties.

AXIS-Stream decompression core

Version

Version: 2.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used
and copied only in accordance with the terms of such
license and with the inclusion of the above copyright
notice.

This software or any other copies thereof may not be
provided or otherwise made available to any other person.
No title to and ownership of the software is hereby
transferred.

The information in this software is subject to change
without notice and should not be construed as a commitment
by the holder of the copyright.

If the Software contains encrypted files you are not allowed to
de-crypt or obtain any encrypted information of the delivered
files. Any publication or making the source code available
to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and
publication and will be held liable for any damages caused by
violations of the above rules.

Module

```
module nc16fc_decompress128_v1 #  
(  
  
    // Users to add parameters here  
  
    // User parameters ends  
  
    // Do not modify the parameters beyond this line  
  
    // Parameters of Axi Master Bus Interface M00_AXIS  
  
    // Parameters of Axi Slave Bus Interface S00_AXIS  
  
)  
(  
    // Users to add ports here  
  
    input wire      cclk,  
    output wire     status,  
  
    // User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

Decompresses the stream with two 16bit words at each sample into a stream which was decompressed by nc16fc_compress128_v1

The module needs two AXIS streams which do not honor TKEEP/TSTRB signals. All Input and outputs are 32bit wide.

Input is received from axis stream interface s00_axis_*
Output is send to axis stream interface m00_axis_*

The core needs 2 resets and clocks for the AXIS in/out busses. The clocks must have the same speed. Different clock frequencies are reserved for future use. Resets should be used also the same for the two buses.

Also required for making the core more stable against timing issues is to provide a compute clock (cclk) which must have exactly half of the AXIS clock frequency.

The clock can be connected to the AXIS clocks too but then the total core speed is reduced to 50% (around 80Mhz on Zynq7K)

The timing of the core is:

- 1.) always N words ($N \leq 65$ for one compressed bock) which belongs to a block of 128 samples are read in on input.
AXIS/TREADY is hold high during that time.
- 2.) after the N words TREADY is deasserted until clock cycle 65
- 3.) output starts always after $128 + x$ cycles
(x is a small number depending on the implementation)
- 4.) after receiving TLAST the last samples
are send to the output with the AXIS/TLAST flag set at the last word.

The decompression core presents the data the same way as nc16fc_compress128_v1 has received the data.

NOTE: all samples (1-16bit half words) must be presented to bit 0-15 and 16-31.
E.g. two 12bit samples should be applied to bit 0-11 (first sample) and
bit 12-27 (second sample). Otherwise the compressor fail to compress the data.

The status bit is currently unused and is connected to 0.
The core can not produce any errors, because all input data can be
compressed or packed. The signal is reserved for future use.
If invalid compression data is send to the core the output delivers
wrong data, but the last flag is still terminating a frame of data.
Thus resynchronization should be possible.

Constraints

The core does not need and special constraints.

If you are using it for less than 16bit wide samples it is better
disconnect the unused bits on output. This ensures that the unused logic
is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper
the name of the module is then

```
nc16fc_decompress128
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements.
You may also include Xilinx specific properties.

AXIS-Stream decompression core

Version

Version: 1.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice.

This software or any other copies thereof may not be provided or otherwise made available to any other person. No title to and ownership of the software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by the holder of the copyright.

If the Software contains encrypted files you are not allowed to de-crypt or obtain any encrypted information of the delivered files. Any publication or making the source code available to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and publication and will be held liable for any damages caused by violations of the above rules.

Module

```
module nc16fc_decompress64_v2 #
(

// Users to add parameters here

// User parameters ends

// Do not modify the parameters beyond this line

// Parameters of Axi Master Bus Interface M00_AXIS

// Parameters of Axi Slave Bus Interface S00_AXIS

)
(
// Users to add ports here

input wire      cclk,
output wire     status,

// User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

Decompresses the stream with two 16bit words at each sample into a stream which was decompressed by nc16fc_compress64_v2

The module needs two AXIS streams which do not honor TKEEP/TSTRB signals. All Input and outputs are 32bit wide.

Input is received from axis stream interface s00_axis_*
Output is send to axis stream interface m00_axis_*

The core needs 2 resets and clocks for the AXIS in/out buses. The clocks must have the same speed. Different clock frequencies are reserved for future use. Resets should be the same for the two buses. Resets must be synchronous to the clocks supplied.

Also required for making the core more stable against timing issues is to provide a compute clock (cclk) which must have exactly half of the AXIS clock frequency.

The clock can be connected to the AXIS clocks too but then the total core speed is reduced to 50% (around 80Mhz on Zynq7K)

The timing of the core is:

- 1.) always N words (N<=33 for one compressed block) which belongs to a block of 64 samples are read in on input.
AXIS/TREADY is hold high during that time.
- 2.) after the N words TREADY is deasserted until clock cycle 33
- 3.) output starts always after 64 + x cycles
(x is a small number depending on the implementation)
- 4.) after receiving TLAST the last samples
are send to the output with the AXIS/TLAST flag set at the last word.

The decompression core presents the data the same way as nc16fc_compress64_v2 has received the data.

The status bit is currently unused and is connected to 0.
The core can not produce any errors, because all input data can be compressed or packed. The signal is reserved for future use.
If invalid compression data is send to the core the output delivers wrong data, but the last flag is still terminating a frame of data.
Thus resynchronization should be possible.

Constraints

The core does not need and special constraints.

If you are using it for less than 16bit wide samples it is better disconnect the unused bits on output. This ensures that the unused logic is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper
the name of the module is then

```
nc16fc_decompress
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements.
You may also include Xilinx specific properties.

AXIS-Stream decompression core

Module

```
module nc16fc_pack128_bayer12bit #
(

// Users to add parameters here

// User parameters ends

// Do not modify the parameters beyond this line

// Parameters of Axi Master Bus Interface M00_AXIS

// Parameters of Axi Slave Bus Interface S00_AXIS

)
(
// Users to add ports here

input wire      cclk,
output wire     status,

// User ports ends

// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

reads 2 x 12 bit data from compressor and outputs
mipi csi-2 32bit data stream

AXIS-Stream decompression core

Module

```
module nc16fc_decompress64_bayer12bit #
(

    // Users to add parameters here

    // User parameters ends

    // Do not modify the parameters beyond this line

    // Parameters of Axi Master Bus Interface M00_AXIS

    // Parameters of Axi Slave Bus Interface S00_AXIS

)
(
    // Users to add ports here

    input wire      cclk,
    output wire     status,

    // User ports ends
    // Do not modify the ports beyond this line

    // Ports of Axi Master Bus Interface M00_AXIS
    input wire      m00_axis_aclk,
    input wire      m00_axis_aresetn,
    output wire     m00_axis_tvalid,
    output wire     [31: 0] m00_axis_tdata,
    output wire     m00_axis_tlast,
    input wire      m00_axis_tready,

    // Ports of Axi Slave Bus Interface S00_AXIS
    input wire      s00_axis_aclk,
    input wire      s00_axis_aresetn,
    output wire     s00_axis_tready,
    input wire      [31 : 0] s00_axis_tdata,
    input wire      s00_axis_tlast,
    input wire      s00_axis_tvalid

);
```

Description

bayer 32bit packed data decompressor

this module takes 32bit compressed data from nc16fc_compress64_v2 read by AXIS s00, decompresses them via nc16fc_decompress64_v2 and pack them to bayer 12bit packed data send to AXIS M00

AXIS-Stream decompression core

Version

Version: 2.0
Date: 2021

Copyright

Copyright (C) 2021 by MIZZI Computer Software GmbH,
Thomas Kihm, Germany.

This software is furnished under a license and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice.

This software or any other copies thereof may not be provided or otherwise made available to any other person. No title to and ownership of the software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by the holder of the copyright.

If the Software contains encrypted files you are not allowed to de-crypt or obtain any encrypted information of the delivered files. Any publication or making the source code available to any other person or company is not allowed.

You are responsible for any damage caused by de-cryption and publication and will be held liable for any damages caused by violations of the above rules.

Module

```
module nc16fc_decompress64_pt_v2 #  
(  
  
    // Users to add parameters here  
  
    // User parameters ends  
  
    // Do not modify the parameters beyond this line  
  
    // Parameters of Axi Master Bus Interface M00_AXIS  
  
    // Parameters of Axi Slave Bus Interface S00_AXIS  
  
)  
(  
    // Users to add ports here  
  
    input wire      cclk,  
    output wire     status,  
  
    // User ports ends
```

```
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS

input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS

input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

A test core which pass through all data without compression.
all timing is the the same as for (de)compress64.

For test only!

Constraints

The core does not need and special constraints.

If you are using it for less than 16bit wide samples it is better
disconnect the unused bits on output. This ensures that the unused logic
is removed from the implementation.

VHDL Wrapper

The module can be called via a VHDL wrapper
the name of the module is then

```
nc16fc_decompress
```

because the wrapper does not support versioning

The wrapper can be modified to meet any of your requirements.
You may also include Xilinx specific properties.

AXIS-Stream compression core

Module

```
module nc16fc_compress_pt #
(

// Users to add parameters here

// User parameters ends

// Do not modify the parameters beyond this line

// Parameters of Axi Master Bus Interface M00_AXIS

// Parameters of Axi Slave Bus Interface S00_AXIS

)
(
// Users to add ports here

input wire      cclk,
output wire     status,

// User ports ends
// Do not modify the ports beyond this line

// Ports of Axi Master Bus Interface M00_AXIS
input wire      m00_axis_aclk,
input wire      m00_axis_aresetn,
output wire     m00_axis_tvalid,
output wire     [31: 0] m00_axis_tdata,
output wire     m00_axis_tlast,
input wire      m00_axis_tready,

// Ports of Axi Slave Bus Interface S00_AXIS
input wire      s00_axis_aclk,
input wire      s00_axis_aresetn,
output wire     s00_axis_tready,
input wire      [31 : 0] s00_axis_tdata,
input wire      s00_axis_tlast,
input wire      s00_axis_tvalid

);
```

Description

passthru stream data from axi stream s00_* to stream data m00_*
for test only. compressor and decompressor are the same module

input from axis stream interface s00_axis_*
output send to axis stream interface m00_axis_*

AXIS-Stream decompression core

Module

```
module nc16fc_decompress_pt #
(

    // Users to add parameters here

    // User parameters ends

    // Do not modify the parameters beyond this line

    // Parameters of Axi Master Bus Interface M00_AXIS

    // Parameters of Axi Slave Bus Interface S00_AXIS

)
(
    // Users to add ports here

    input wire      cclk,
    output wire     status,

    // User ports ends
    // Do not modify the ports beyond this line

    // Ports of Axi Master Bus Interface M00_AXIS
    input wire      m00_axis_aclk,
    input wire      m00_axis_aresetn,
    output wire     m00_axis_tvalid,
    output wire     [31: 0] m00_axis_tdata,
    output wire     m00_axis_tlast,
    input wire      m00_axis_tready,

    // Ports of Axi Slave Bus Interface S00_AXIS
    input wire      s00_axis_aclk,
    input wire      s00_axis_aresetn,
    output wire     s00_axis_tready,
    input wire      [31 : 0] s00_axis_tdata,
    input wire      s00_axis_tlast,
    input wire      s00_axis_tvalid

);
```

Description

passthru stream data from axi stream s00_* to stream data m00_*
for test only. compressor and decompressor are the same module

input from axis stream interface s00_axis_*
output send to axis stream interface m00_axis_*