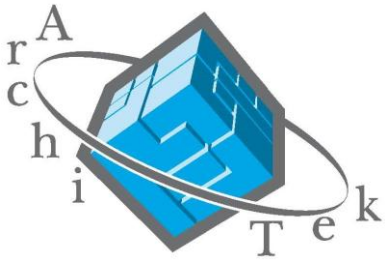


frBLT Specification

Frame Bit Block Transfer

Revision 1.5

10 April 2025

English Edition

1. Overview

1.1. Introduction

- Frame Bit Block Transfer (hereinafter referred to as frBLT) is a compact transfer engine that transfers image data from a source image to a destination image. Up to two mapping operations can be applied in the transfer path between the source and destination.
- Supported pixel formats include 8-bit x 4 components (32-bit), 8-bit x 3 components (24-bit), and RGB565/YUYV (16-bit).
- Coordinates are provided to derive source and destination positions, and transfer is performed from the source image to the destination image. Images can be processed fragmentarily by splitting into several lines. Each fragment can have a different context (image processing task), and performance will not degrade. By processing multiple tasks in a time-division manner, simultaneous processing can be achieved in appearance.
- Source coordinates can be mapped using mapping data stored in memory. No interpolation methods such as bi-linear interpolation are supported. Since this is a simple transformation using the loaded data as an address, it allows for broad applications, but performance can degrade if discrete indexing occurs frequently.
- Mapping is parallelized into two stages: one for RGB and one for Alpha, making it easy to apply for color conversion.
- It supports simultaneous input of up to four planes and allows packing into arbitrary formats for output (combination mode).

1.2. Main Parameters

- **Memory Bus**: Pixel Read/Write: 64-bit x 1
  Command List Read: 64-bit x 1
- **Throughput**: Maximum 1 pixel per cycle
- **Pixel Formats**:
  - 1-bit component
  - 8-bit component (grayscale)
  - 16-bit component (RGB565, ARGB1555, YUV422)

    - 24-bit component (RGB888, YUV, etc.)
    - 32-bit component (ARGB8888, AYUV, etc.)
- **Clock**: Undefined (depends on implementation process)

1.3. Implementation Parameters

Parameter Name	Description	Default Value
BLR	• Radix of burst length for command list reading • Sets the burst size for 64-bit memory access	1 (up to 4)
BSR	• Radix of burst length for data read/write • Sets the burst size for 64-bit memory access	2 (up to 4)

1.4. Others

- The **ItalicBold** font represents *cores*.
- The **Thoma** font represents *signals*.
- The **Command.Field** font represents *command list names and field names*. Field names may be omitted in some cases.

2. Signal Lines

2.1. Control Bus Interface

Signal Name	IO	Pol	Source	Description
cntlReq	I	+	clk	• Request signal • Evaluate cntlGnt
cntlGnt	O	+	clk	• Grant signal
cntlRxw	I	+	clk	• R/W signal • Evaluate cntlReq & cntlGnt 0: Write 1: Read
cntlAddr[31:0]	I	+	clk	• Address signal • Evaluate cntlReq & cntlGnt

cntlWrAck	O	+	clk	• Writ acknowledge signal
cntlWrData[31:0]	I	+	clk	• Write data signal • Evaluate cntlWrAck
cntlRdAck	O	+	clk	• Read acknowledge signal
cntlRdData[31:0]	O	+	clk	• Read data signal • Sync cntlRdAck
cntlIrq	O	+	clk	• Interrupt signal • Level hold type(Fix'0')

2.2. PSS Interface

Signal Name	IO	Pol	Source	Description
iVld	I	+	clk	• Pipeline start valid signal
iStall	O	+	clk	• Pipeline start stall signal
iAddr[31:0]	I	+	clk	• Address to fetch context data • Evaluate iVld & !iStall
iDelta[15:0]	I	+	clk	• Transfer volume • Evaluate iVld & !iStall
iIndex[64:0]	I	+	clk	• Five coordinates to specify the processing • Evaluate iVld & !iStall
oVld	O	+	clk	• Pipeline end valid signal
oStall	I	+	clk	• Pipeline end stall signal

2.3. Memory Interface (Pixel R/W Use)

Signal Name	IO	Pol	Source	Description
miReq	O	+	clk	• Request signal
miGnt	I	+	clk	• Grant signal
miRxw	I	+	clk	• R/W signal • Write indicates cache flush
miAddr[31:0]	O	+	clk	• Address signal
miWrStrb	O	+	clk	• Write strobe
miWrAck	I	+	clk	• Write acknowledge signal

miWrData[63:0]	O	+	clk	• Write data signal
miWrMask[7:0]	O	+	clk	• Write mask signal
miRdStrb	O	+	clk	• Read strobe
miRdAck	I	+	clk	• Read acknowledge signal
miRdData[63:0]	I	+	clk	• Read data signal

2.4. Memory Interface (Parameter Read Use)

Signal Name	IO	Pol	Source	Description
meReq	O	+	clk	• Request signal
meGnt	I	+	clk	• Grant signal
meAddr[31:0]	O	+	clk	• Address signal
meStrb	O	+	clk	• Read strobe signal
meAck	I	+	clk	• Read acknowledge signal
meFlush	O	+	clk	• Read flush signal
meData[63:0]	I	+	clk	• Read data signal

2.5. Utility

Signal Name	IO	Pol	Source	Description
reg_swap	O	+	clk	• 64bit swap signal
rstReq	O	+	clk	• Internal reset signal to reset the external system
rstAck	I	+	clk	• Acknowledge of rstReq
fReq	I	+	clk	• 1 clock early request against the miReq signal • Use to generate gate signal (for mc2)
pReq	O	+	clk	• 1 clock early request against the meReq signal • Use to generate gate signal (for mc2)
gate[7:0]	O	+	clk	• Gated clock control signal signifying condition of each internal block
gclk[7:0]	I	+	clk	• Gated clock
clk	I	+	clk	• Clock

reset	I	+	clk	• Synchronous reset signal

3. Architecture and Operation Description

3.1. Architectural Overview

- The **Pipeline Slice Scheduler (hereinafter referred to as pss)** retrieves the required context from memory, fragments the information, generates coordinate data, and then activates the **frBLT**. For more details about **pss**, please refer to its specification document.
- The connection interface only requires the coordinates and the start address of the command list to be provided. Since control is based on a simple **Valid/Stall** mechanism, the use of **pss** is not mandatory. In such cases, the **pss** block may be replaced by a custom core.
- The **frBLT** follows a pipeline structure as shown in **Figure 1**, and processing proceeds in the order: **Initiator** → **Read** → **Remap** → **Clut/Mlut** → **Write**.

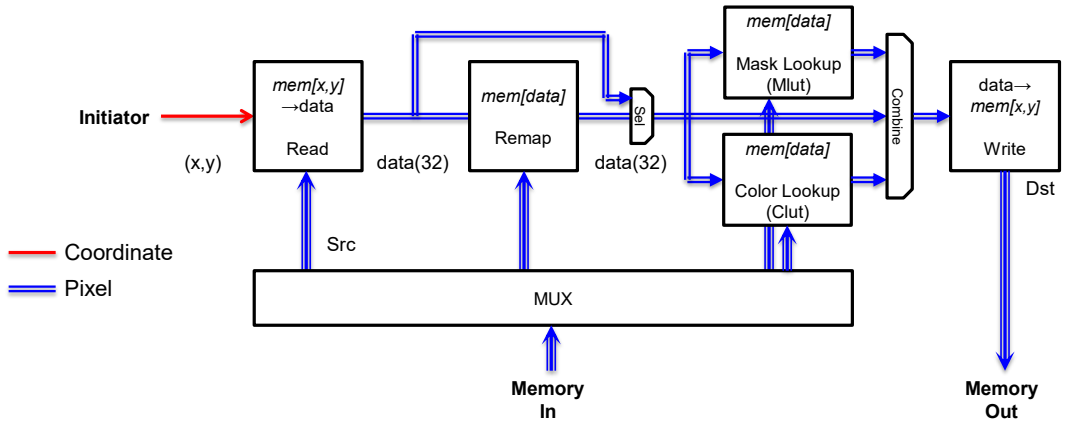
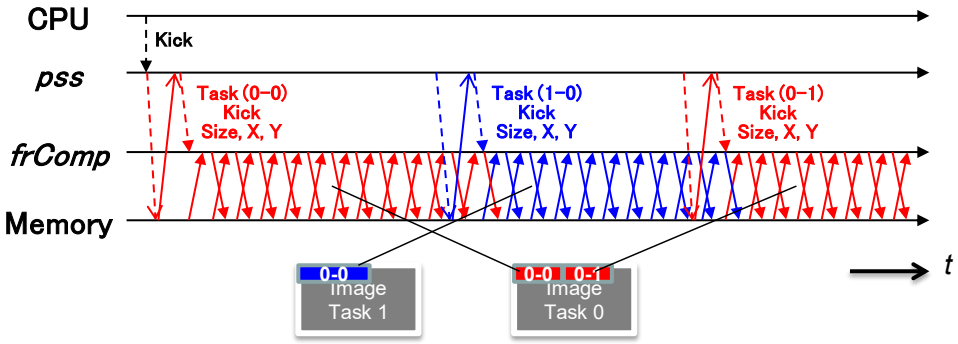


Figure 1 *frBLT* Block Diagram

3.2. Drive Interface (Initiator)

- The **pss** scans the intermediate coordinates (used as indices for the final coordinates) along the X-axis and sends them to the **Initiator** of the **frBLT**. Configuration data for the **pss**—such as image information and processing units—should be preloaded into memory.
- The **pss** manages multiple configurations (denoted as N, which varies depending on the implementation) in a time-multiplexed manner. After scheduling, it drives the **frBLT** accordingly.



- The **Initiator** reads the command list from memory based on the image information sent by **pss**, and sets up the pipeline. The parameters extracted from the command list are managed with a triple-buffering mechanism, so unless the specified processing unit is extremely short, there is no performance degradation. Even if the unit is short, as long as the same context continues, the system performs concatenated processing, avoiding performance loss.
- When **pss** is not used, a basic method for driving the **frBLT** is as follows (executed per line):
    - Prepare a counter to count from 0 to (height - 1) in the Y-direction, assert the iVld signal when valid, and increment the counter when iStall is '0'.
    - Assert the start address of the 32-byte command list to the iAddr signal.
    - Assert the width in the X-direction minus one to the iDelta signal.
    - Assert the counter value to iIndex[31:16]; other iIndex signals should be asserted with '0'.

3.3. Fragmentation Considerations

- When fragmenting processing, alternating between different parameters generally does not cause inconsistencies. However, when using the mapping function, be cautious when modifying the mapping data, since it references memory for such data.

3.4. Transfer

- A simple transfer is performed by configuring the read from the source (Src) and the write to the destination (Dst). While the transfer size must be the same for both, the format and stride can be set independently.
- 2D block transfer is supported. The image below illustrates how the transfer works:
  Data corresponding to the format setting is processed based on the base address, incremented by the X-coordinate. When the Y-coordinate is updated, (Stride + 1) × format amount is added to the address. t
  Thevalues for X and Y coordinates are provided by **pss**.

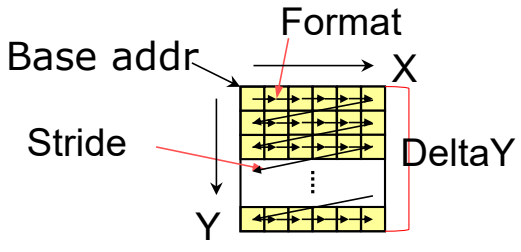


Figure 2 2D Block Transger

 It is possible to **pack n-bit data into 1-bit data**. The transfer size is generally adjusted to ensure that the write access is aligned to 32-bit. The packing direction can be selected as either **LSB-first** or **MSB-first**.

 4-dimensional block transfer is supported. A 2D block defined by the X and Y coordinates can be extended in the horizontal (W) and vertical (Z) directions. The diagram below illustrates this transfer method:

- Based on the **base address**, data corresponding to the format setting is processed incrementally along the X coordinate.
- When the **Y coordinate** is updated, an address increment of (Stride + 1) × format is applied.
- For **4D transfers**, additional address offsets are added when the **Z coordinate** changes (using **StrideZ**) and when the **W coordinate** changes (using **StrideW**).
- Just like X and Y, the values for W and Z coordinates are also updated and supplied by the **pss**.

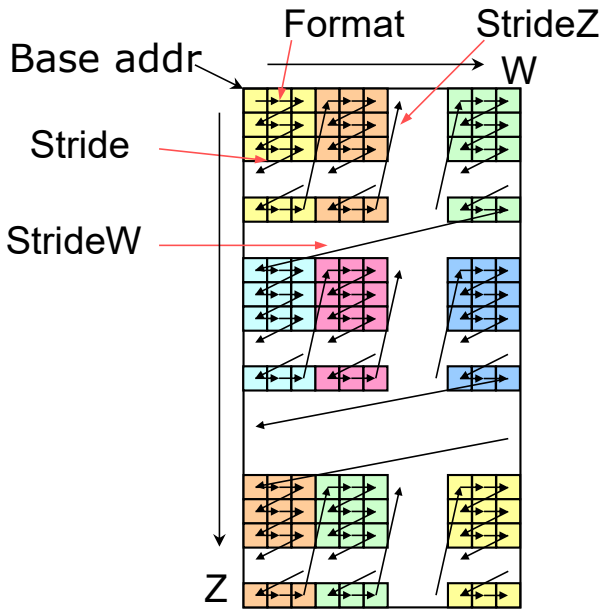


Figure 3 4D Block Transger

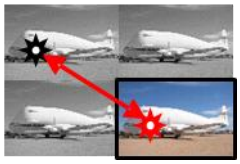
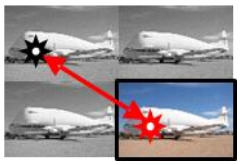
3.5. Transfer Volume Expansion

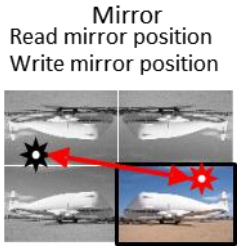
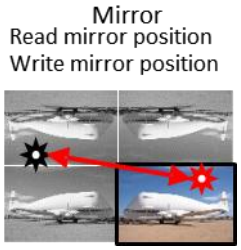
- When the **source format is 32 bpp**, the transfer volume can be expanded up to **128 times**. This expansion also applies to the **destination**. Transfers exceeding **65536 × 32bpp** can be handled in a single operation.
- When using expanded transfer volume, **all options such as edge handling, offset, and mapping must be disabled**.

3.6. Data Reading

- It is possible to perform a **fill operation** where the source coordinates are fixed or the source is not referenced at all.
    - In the first case, the same data from memory is repeatedly read. Due to internal caching, memory access only occurs at the initial point.
    - In the second case, **no memory access is performed**, and a fixed value is provided instead.
- When **Mlut mapping** is used, the fixed value is **0**. Otherwise, the **Mlut control value** is used as a substitute.
- When **4D transfer** is used, the fixed value is also **0**; otherwise, the **Mlut control value** is substituted.
- You can define **boundaries for the X and Y coordinates**, and configure how out-of-bound (OOB) access is handled.
- On the **Beppu platform**, SrcInfo.Stride = 0 posed no problem when reading **Clut data**, but on **Chichibu**, proper Clut data reading required a valid SrcInfo.Stride setting.

Edge	Description
0 CV: Exclusive	"If the center coordinate is out of bounds, all pixel cache values are replaced with the default value.
1 CV: -	No out-of-bounds check is performed.
8	Out-of-bounds pixels are replaced with the

CV: Constant	Default Read 'default(C2)' value Write masking  default value.
9 CV: Replica	Copy Read nearest pixel Write nearest pixel  Out-of-bounds pixels are replaced with the nearest neighbor pixel value.
value10 CV: Warp	Ring Read repeat pixel Write repeat pixel  Out-of-bounds pixels are replaced with the corresponding value from a wrap-around image.
10 CV: Warp	Ring Read repeat pixel Write repeat pixel  Out-of-bounds pixels are replaced with the corresponding value from a wrap-around image.

11 CV: Reflect_101	 Out-of-bounds pixels are replaced with the corresponding value from a mirrored image (pixels beyond the edge reflect back from the edge point).
15 CV:Reflect	 Out-of-bounds pixels are replaced with the corresponding value from a mirrored image (pixels beyond the edge repeat the edge point itself).

3.7. Transposition

X and Y coordinates can be swapped to perform transposition. Both the source and destination can be targets of this operation.

SrcX = SrcSwapX ? Y : X
SrcY = SrcSwapY ? X : Y

DstX = DstSwapX ? Y : X
DstY = DstSwapY ? X : Y

Combining this with boundary handling allows operations such as horizontal flipping and 90° rotation.

3.8. Offset

Offsets can be set for the X and Y coordinates of both the source and destination. However, if the mapping function is used, some or all of the offsets may be set to

0.
Likewise, when using 4D transfer, some or all offset values must also be 0.

3.9. Mapping

Mapping is a general term for data transformation, including coordinate and color conversion. It can be performed using any one or a combination of the following: Remap, Clut, and Mlut.

Unlike Read/Write operations, Remap, Clut, and Mlut use simple 1D addressing. Each performs memory access by adding input data to its base address.
The type of data added to the address can be 1, 8, 16, or 32 bits.

To perform coordinate mapping, a list of coordinates must be prepared in memory. A 2D Read operation loads these values, which are then used as new coordinates in the Remap operation. The data used for Read must be 1D converted beforehand.
Example:
- Assume 1280×720 32-bit pixel data
- Sequentially read new coordinates
- Coordinates must be 1D converted beforehand (1280×Y + X)
- Access data with Remap using 1280×Y + X

Although arbitrary mapping is possible, there are two limitations: interpolation (e.g., bilinear) is not supported (nearest-neighbor only), and one mapping entry is required for each pixel, which increases data volume. For 24-bit RGB conversion, a 64 MB mapping table is required (2^{24} × 4 bytes).

Remap and Clut are functionally equivalent and can output 1, 8, 16, or 32-bit data. Mlut is limited to 1 or 8-bit output. Clut and Mlut operate in parallel and their results can be merged. They can also be used for write masks.

3.10. Element Combination

Up to 4 arbitrary elements from different images can be combined into a single image. Up to four independently formatted images can be assigned to Src, Remap, Clut, and Mlut. Specific elements from each can then be merged. The

size and stride follow the Src settings. This feature is exclusive of the mapping functions.

Remap, Clut, and Mlut generate destination pixel components based on their byte-wise shift values and formats. Src has no shift value. Note that overlapping elements may result depending on the shift and format settings. In such cases, the overlapping data will be OR'ed.

Elements selected by Clut or Mlut can be used either as pixel values or as masks for the destination. Only one of these options can be selected at a time.

3.11. Data Writing

It is possible to configure the engine so that no memory write accesses are performed. This allows prefetching for the memory system by only performing reads. Example settings:
- Configure 1D access in pss (number of cache lines to prefetch)
- SrcSwapY = 1 (substitute Y with X coordinate changes)
- SrcStride = cache line length - 1
- DstDisable = 1 (disable writing)

It is also possible to configure masked writes to memory, enabling cache flushes. Example settings:
- Configure 1D access in pss (number of cache lines to flush)
- SrcScan = fixed value reference (no read access)
- DstSwapY = 1 (substitute Y with X coordinate changes)
- DstOffsetY = cache line length - 1 (only for last entry in cache line)
- DstStride = cache line length - 1
- DstMask = 1 (enable write mask)

3.12. Connection with *pss*

The address output from pss (iAddr signal) is used to fetch the command list from memory. Refer to the command list section for details. If pss is not used, directly access the pss interface.

Using the coordinate output (iIndex signal) and parameters from the command

list, the base addresses for the input and output image data are calculated. The formula is as follows:
Address = Base + Format × (X + Stride × Y + StrideZ × Z + StrideW × W)

A scanline process is carried out for the length indicated by the iDelta signal, starting from coordinates X and Y. The iDelta signal effectively defines the fragment size. Line ends may result in non-aligned (partial) segments. Full-line processing is also supported.

3.13. Performance

The engine processes one pixel per clock cycle. The number of elements per pixel does not affect this performance.

3.14. Restrictions

- SrcOffset and Remap cannot be enabled at the same time.
- DstOffset and Clut cannot be enabled simultaneously.
- Boundary fixed value and Mlut cannot be used together.
- 4D transfer and Mlut cannot be used simultaneously.
- The stride for input/output address updates is limited to a maximum of 0xffff.

4. Register Description

4.1. Overview

Registers are accessed via the control bus.

As some registers affect the pipeline's behavior and performance, timing of their configuration must be handled with care.

Access Types:
R - Read Only (Write has no effect)
R/W - Read / Write
R/WC - Read / Write Clear

Do not access reserved registers. For reserved fields, always write '0'.
Addresses or data marked with 'x' are 'don't care' values.

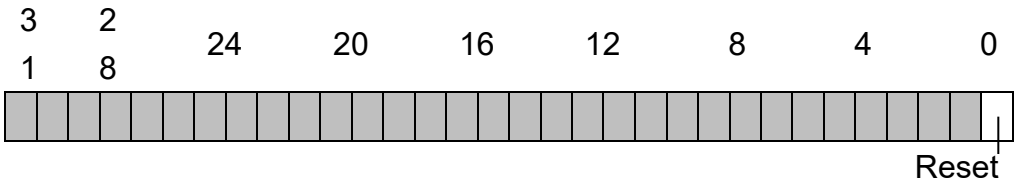
4.2. Register Definitions

Address	Register Name	Description
0000_0000	Reset	リセット制御
0000_0004	System	システム制御

4.3. Details

4.3.1.1. Reset Register

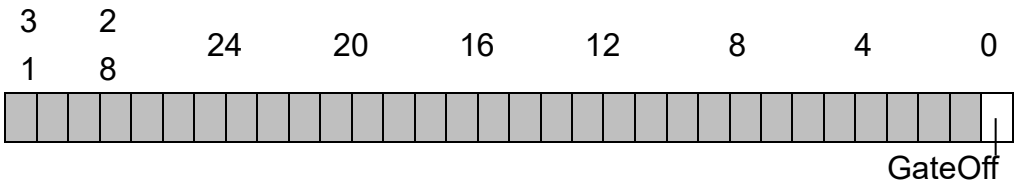
[Address: 0x0000_0000]



Name	Type	Default	Description
			Synchronous reset. Setting this field to '1' places the system in a reset state, which is then automatically cleared. Unlike the reset_n signal, the contents of other registers are preserved.
Reset	R/W	0	When set to '1', the rstReq signal is immediately asserted. This signal notifies external logic that the frBLT is in a reset state and requests handling. Once the handling is complete, the rstAck signal must be asserted (if handling is unnecessary, rstAck should always be held at '1'). After this handshake is complete, the Reset field will automatically return to '0'.

4.3.1.2. System Register

[Address: 0x0000_0004]



Name	Type	Default	Description
GateOff	R/W	0	Gated Clock Off Mode. When set to '1', all bits of the gate signal are fixed to '1'.

5. Command List

5.1. Overview

- The **command list** is stored in memory in 32-byte units. The start address of the command list is indicated by the iAddr signal output from the **pss**. After initialization, **frBLT** fetches the command list and stores the values in internal registers.
- Each pipeline stage manages the required parameters independently, in synchronization with its processing timing. This design enables **seamless execution of different command lists**. There is **no need for synchronization processing**, such as monitoring completion from the **pss** side.
- For all **reserved commands and fields**, set their values to **'0'**.
- The addresses described are **relative to the value output by pss**, and must be aligned on **16-byte boundaries**.

5.2. Definition

Address	Command Name	Description
00	MasterCntl	Master Control

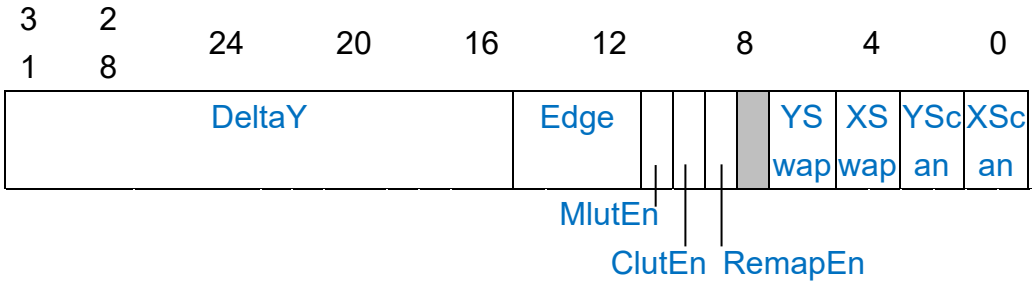
04	RemapCntl	Remap Control
08	ClutCntl	Clut Control
0c	MlutCntl	Mlut Control
10	SrcInfo	Source information
14	SrcBase	Source base address
18	DstInfo	Destination information
1c	DstBase	Destination address

5.3. Details

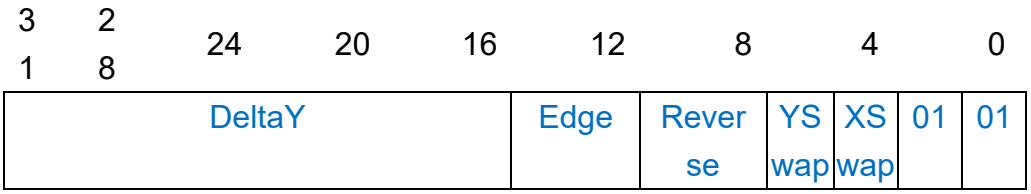
5.3.1.1. MasterCntl Command

[Address: 0x00]

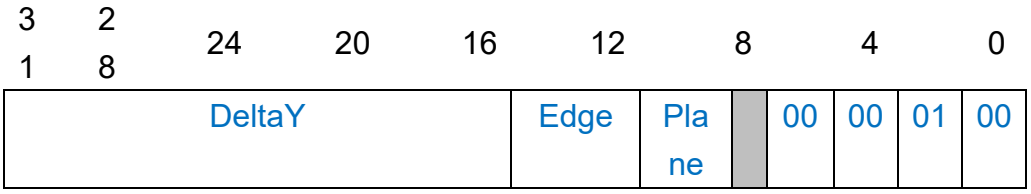
Normal Mode:



4 Dimensional Mode:



Concatenation Mode:



Name	Description
DeltaY[15:0]	Specifies the transfer size as ΔY - 1. This field is used when configuring boundaries. If boundary configuration is

not used, this field may be set to Unknown. In 4-dimensional mode, it also defines the width along the Y-axis.

Edge[3:0] Configures the endpoint option for acquiring input data.

Edge	Description
0	Replaces out-of-bound areas with a fixed value.
1	Does not perform out-of-bound region checking.
2-7	Reserved
8	Out-of-bound areas are replaced with a fixed value.
9	Out-of-bound areas are replaced with the nearest valid value.
10	Out-of-bound areas are replaced with the value indicated by the circular (wraparound) coordinates.
11	Out-of-bound areas are replaced with the value indicated by the mirrored coordinates (without duplication at the reflection boundary).
12-14	Reserved
15	Out-of-bound areas are replaced with the value indicated by the mirrored coordinates (with duplication at the reflection boundary).

MlutEn / Plane[3] Enables the Mlut function in normal mode when set to '1'. This setting is prohibited when fixed values are selected for YScan = 2 and XScan = 2.

ClutEn / Plane[2] Enables the Clut function in normal mode when set to '1'. This setting is prohibited when DstOffset is used.

RemapEn / Plane[1] Enables the Remap function in normal mode when set to '1'. This setting is prohibited when SrcOffset is used.

Reverse[3:0] Specifies the reverse scanning direction for each axis in 4-dimensional mode. The LSB corresponds to the X-axis, and the MSB corresponds to the W-axis. Setting a bit to '1' enables scanning from the end point to the start point along the corresponding axis. A separate width setting is required to define the end point. The width for each axis is specified as follows: iDelta signal for the X-axis, MasterCntl.WidthY for the Y-axis, Delta.WidthZ for the Z-axis, and Delta.WidthW for the W-axis.

YSwap[1:0] Specifies the input for the Y-axis. Set to '00' when in concatenation mode.

YSwap[0]	Description
0	Assigns an internally generated Y value to the input coordinate Y.
1	Assigns an internally generated X value to the input coordinate X.

YSwap[1]	Description
0	Assigns an internally generated Y value to the output coordinate Y.
1	Assigns an internally generated X value to the output coordinate X.

XSwap[1:0] Specifies the input for the X-axis. Set to '00' when in concatenation mode.

XSwap[0]	Description
0	Assigns an internally generated X value to the input coordinate X.
1	Assigns an internally generated Y value to

	the input coordinate Y.

XSwap[1]	Description
0	Assigns an internally generated X value to the output coordinate X.
1	Assigns an internally generated Y value to the output coordinate Y.

YScan[1:0] Configures the scanning behavior in the Y-axis direction. Set to '01' in both 4-dimensional mode and concatenation mode.

YScan	Description
0	Forward scanning (incrementing coordinate Y from 0).
1	Selects between 4-dimensional mode and concatenation mode. *Mode selection is determined by the value of XScan.*
2	Fixes coordinate Y to 0.
3	Performs no coordinate operations on the Source input and uses SrcBase as a fixed value (Fill). *Note: XScan must also be set accordingly.*

XScan[1:0] Configures the scanning behavior in the X-axis direction. Set to '01' in 4-dimensional mode and '00' in concatenation mode.

XScan	Description
0	Forward scanning (incrementing coordinate X from 0). *Concatenation mode is selected when YScan is set to '01'.*
1	Selects 4-dimensional mode.

	Note: YScan must also be set accordingly.
2	Fixes coordinate X to 0.
3	Performs no coordinate operations on the Source input and uses SrcBase as a fixed value (Fill). *Note: YScan must also be set accordingly.*

5.3.1.2. RemapCntl / Plane1 / SrcOffset / Delta Command

[Address: 0x04]

Normal Mode（MasterCntl.RemapEn=1）/ Concatenation Mode： RemapCntl / Plane1

31	28	24	20	16	12	8	4	0
Addr						Opt	Shift	Format

Normal Mode（MasterCntl.RemapEn=0）： SrcOffset

31	28	24	20	16	12	8	4	0
OffsetY						OffsetX		

4 Dimensional Mode： Delta

31	28	24	20	16	12	8	4	0
WidthW						WidthZ		

Name	Description
Addr[31:8]	Specifies the upper address for Remap / Plane1. This setting is valid only when MasterCntl.RemapEn is set to 1.
Opt[1:0]	Set to 0 (fixed)
Shift[2:0]	Specifies the byte-level shift amount for Remap / Plane1

using two's complement representation (left shift is positive, right shift is negative). After shifting, the number of bytes corresponding to the Format is allocated. This setting is valid in concatenation mode when MasterCntl.Plane[1] = 1.

Format[2:0] Specifies the format for Remap / Plane1. This setting is valid when both MasterCntl.RemapEn and MasterCntl.Plane[1] are set to 1.

Format	Description	Note
0	8Bpp	The upper 24 bits of the resulting 32-bit data are set to 0.
1	16Bpp	The upper 16 bits of the resulting 32-bit data are set to 0.
2	24Bpp	-
3	32Bpp	-
4	1Bpp	The result is generated by copying one bit at a time, starting from the most significant bit (MSB). This field is reserved when operating in concatenation mode.
5	1Bpp	The result copies 1 bit in order from the LSB. This field is reserved in concatenation mode.
6	Reserved	-
7	Reserved	-

OffsetY[15:0] SrcOffsetY. This setting is valid when MasterCntl.RemapEn is set to 0.

OffsetX[15:0] SrcOffsetX. This setting is valid when MasterCntl.RemapEn is set to 0.

WidthW[15:0] Specifies the width of the W coordinate used when MasterCntl.Reverse[3] is set to 1 in 4-dimensional mode.

WidthZ[15:0] Specifies the width of the Z coordinate used when MasterCntl.Reverse[2] is set to 1 in 4-dimensional mode.

5.3.1.3. ClutCntl / DstOffset / SrcStride / Plane2 Command

[Address: 0x08]

Normal Mode（MasterCntl.ClutEn=1）/Concatenation Mode： ClutCntl / Plane2

31	28	24	20	16	12	8	4	0
Addr						Opt	Shift	Exp / Format

Normal Mode（MasterCntl.ClutEn=0）： DstOffset

31	28	24	20	16	12	8	4	0
OffsetY						OffsetX		

4 Dimensional Mode： SrcStride

31	28	24	20	16	12	8	4	0
StrideW						StrideZ		

Name	Description
Addr[31:8]	Specifies the upper address. This setting is valid when MasterCntl.ClutEn is set to 1.
Opt[1:0]	Set to 0 (fixed)
Shift[2:0]	Specifies the byte-level shift amount for Clut / Plane2 using two's complement representation (left shift is positive, right shift is negative). After shifting, the number of bytes corresponding to the Format is allocated. This setting is valid in concatenation mode when MasterCntl.Plane[2] = 1.

Exp — Specifies whether the result of Clut / Plane2 is used as data or as a mask. This setting is valid in concatenation mode when MasterCntl.Plane[2] = 1.

Exp	Description	Note
0	Data	-
1	Mask	Masks elements with a value of 0 among the selected items.

Format[1:0] — Specifies the format for Clut / Plane2. This setting is valid when both MasterCntl.ClutEn and MasterCntl.Plane[2] are set to 1.

Format	Description	Note
0	8Bpp	The upper 24 bits of the resulting 32-bit data are set to 0.
1	16Bpp	The upper 16 bits of the resulting 32-bit data are set to 0.
2	24Bpp	-
3	32Bpp	-

OffsetY[15:0] — DstOffsetY. This setting is valid when MasterCntl.ClutEn is set to 0.

OffsetX[15:0] — DstOffsetX. This setting is valid when MasterCntl.ClutEn is set to 0.

StrideW[15:0] — In 4-dimensional mode, specifies the W-axis address update stride minus 1 for the input.

StrideZ[15:0] — In 4-dimensional mode, specifies the Z-axis address update stride minus 1 for the input.

5.3.1.4. MlutCntl / Default / DstStride / Plane3 Command

[Address: 0x0c]

Normal Mode（MasterCntl.MlutEn=1）/**Concatenation Mode**

: MlutCntl / Plane3

3 1	2 8	24	20	16	12	8	4	0

Addr	Opt	Shift	Exp	Format

Normal Mode（MasterCntl.MlutEn=0）： Default

31	28	24	20	16	12	8	4	0
Default								

4 Dimensional Mode： DstStride

31	28	24	20	16	12	8	4	0
StrideW					StrideZ			

Name	Description
Addr[31:8]	Specifies the upper address. Valid only when MasterCntl.MlutEn is set to 1.
Opt[1:0]	Set to 0 (fixed)
Shift[2:0]	Specifies the byte-wise shift amount for Mlut/Plane3 in two's complement format (positive for left shift, negative for right shift). After shifting, a number of bytes equal to Format is allocated. Valid only when MasterCntl.Plane[3] is set to 1 in concatenation mode.
Exp	Specifies whether the result of Mlut/Plane3 is used as data or as a mask. Valid only when MasterCntl.Plane[3] is set to 1 in concatenation mode.

Exp	Description	Note
0	Data	-
1	Mask	Masks the selected elements with zero.

Format[1:0]	Specifies the format of Mlut/Plane3. Valid only when both MasterCntl.MlutEn and MasterCntl.Plane[3] are set to 1.

Format	Description	Note
0	8Bpp	The upper 24 bits of the 32-bit result data are set to 0.
1	16Bpp	The upper 16 bits of the 32-bit result data are set to 0.
2	24Bpp	-
3	32Bpp	-

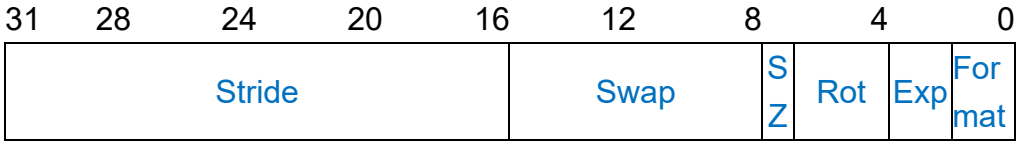
Default[31:0] Sets the default pixel value. Valid only when MasterCntl.MlutEn is 0 and the mode is not 4-dimensional.

StrideW[15:0] Sets the address update stride minus 1 for the W-axis of the output in 4-dimensional mode.

StrideZ[15:0] Sets the address update stride minus 1 for the Z-axis of the output in 4-dimensional mode.

5.3.1.5. SrcInfo Command

[Address: 0x10]



Name	Description
Stride[15:0]	Sets the address update stride minus 1 for the X-axis of the input data. The unit depends on the Format. Not used when MasterCntl.X/YScan is not equal to 3.
Swap[7:0]	Configures the byte swap for the input data. Defines the byte-wise mapping from input data In[31:0] to internal data Pipe[31:0]. If not set as a one-to-one mapping, undefined behavior or data overlap may occur. Not used when MasterCntl.X/YScan is not equal to 3.

Value	Swap[7:6]	Swap[5:4]	Swap[3:2]	Swap[1:0]
	Pipe[31:24	Pipe[23:16	Pipe[15:8]	Pipe[7:0]

	]	]		
0	In[31:24]	In[23:16]	In[15:8]	In[7:0]
1	In[7:0]	In[31:24]	In[23:16]	In[15:8]
2	In[15:8]	In[7:0]	In[31:24]	In[23:16]
3	In[23:16]	In[15:8]	In[7:0]	In[31:24]

SZ

When Stride is 0, use Stride + 1. If set to '0', the value is 0x10000; if set to '1', the value is 0.

Rot[2:0]

In the Pixel Cache, configures the details of the pixel format for the referenced Source input data (see Format for details). Not used when MasterCntl.X/YScan is not equal to 3. When the Format is 16bpp, data lane operations are applied. When the Format is 32bpp, the transfer size is doubled by 2Rot. Increasing the transfer size also affects the transfer size of the output data.

Exp[1:0]

Configures the detailed format of the input data (see Format for details). Not used when MasterCntl.X/YScan is not equal to 3.

Format[1:0]

Sets the Bpp (bits per pixel) of the input data format. Not used when MasterCntl.X/YScan is not equal to 3.

<table>
<tr><th>Format</th><th>Exp</th><th>Pipe [31:24]</th><th>Pipe [23:16]</th><th>Pipe [15:8]</th><th>Pipe [7:0]</th><th>Note</th></tr>
<tr><td rowspan="2">0
8Bpp</td><td>0</td><td>0</td><td>0</td><td>0</td><td>In[7:0]</td><td></td></tr>
<tr><td>1</td><td>In[7:0]</td><td>In[7:0]</td><td>In[7:0]</td><td>In[7:0]</td><td>8bit Replica</td></tr>
<tr><td rowspan="2">0
1Bpp</td><td>2</td><td colspan="4">Internal Special (MSB First)</td><td rowspan="2">MSB First
Only at Ver.A</td></tr>
<tr><td>3</td><td colspan="4">Internal Special (LSB First)</td></tr>
<tr><td rowspan="6">1
16Bpp</td><td>0</td><td>Gray</td><td>In [15:11] [15:13]</td><td>In [10:5] [10:9]</td><td>In [4:0] [4:2]</td><td>RGB565
Lower Replica</td></tr>
<tr><td rowspan="4">1</td><td rowspan="4">In [15:8]</td><td>In[7:0]</td><td>In[7:0]</td><td rowspan="4">In [7:0]</td><td>Rot[1:0]=’0’</td></tr>
<tr><td>In[7:0]</td><td>In[15:8]</td><td>Rot[1:0]=’1’</td></tr>
<tr><td>In[15:8]</td><td>In[7:0]</td><td>Rot[1:0]=’2’</td></tr>
<tr><td>In[15:8]</td><td>In[15:8]</td><td>Rot[1:0]=’3’</td></tr>
<tr><td>2</td><td>0xff</td><td>In</td><td>In</td><td>In</td><td>Alpha=1.0</td></tr>
</table>

			[31:24] /In [15:8]	[23:16]	[7:0]	YUYV
	3	In[31:16] >> Rot		In[15:0] >> Rot		
2 24Bpp	0	In [23:16]	In [23:16]	In [15:8]	In [7:0]	
	1	Gray	In [23:16]	In [15:8]	In [7:0]	
	2	0xff	In [23:16]	In [15:8]	In [7:0]	Alpha=1.0
	3	Gray	Gray	Gray	Gray	All Gray
3 32Bpp	0	In [31:24]	In [23:16]	In [15:8]	In [7:0]	Rot[0]='0'
		8Bpp X[1:0]=0	8Bpp X[1:0]=1	8Bpp X[1:0]=2	8Bpp X[1:0]=3	Rot[0]='1'
	1	Gray	In [23:16]	In [15:8]	In [7:0]	
	2	0xff	In [23:16]	In [15:8]	In [7:0]	Alpha=1.0
	3	Gray	Gray	Gray	Gray	All Gray

In: Memory side
Pipe: Blender side (=ARGB)
Gray: (2 In[23:16] + 5 In[15:8] + In[7:0]) / 8

5.3.1.6. SrcBase Command

[Address: 0x14]

31	28	24	20	16	12	8	4	0
Base[31:6]							Wrap[5:0]	

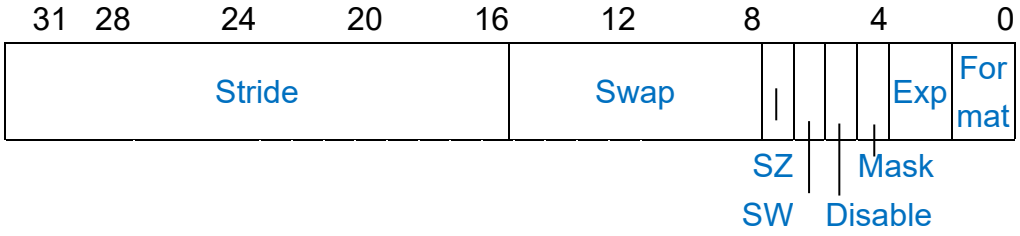
31	28	24	20	16	12	8	4	0
Deafult								

Name	Description
Base[31:6]	Sets the base address of the input data. Must be specified in 64-byte alignment units. Valid only when MasterCntl.X/YScan is not equal to 3.
Wrap[5:0]	By setting the MSB to '1', the 4 bits Wrap[4:1] specify the address mask bits (Cache only). The mask value applied to the 32-bit address is calculated as: 0x007FFFFF >> ~Wrap[4:1]. Additionally, 1 bit (Wrap[0]) is sent as LSB information to the memory system. If the MSB is '0', no masking is applied, and 2 bits (Wrap[1:0]) are sent as LSB information to the memory system.
Deafult[31:0]	Fixed value used for boundary processing. Valid only when MasterCntl.X/YScan is equal to 3.

Wrap[4:1]	Description
0	The lower 8 bits are valid; the upper 24 bits are set to 0.
1	The lower 9 bits are valid; the upper 23 bits are set to 0.
15	The lower 23 bits are valid; the upper 9 bits are set to 0.

5.3.1.7. DstInfo Command

[Address: 0x18]



Name	Description
Stride[15:0]	Sets the address update stride minus 1 for the output data (see also SrcInfo). Note that 0xFFFF is treated as '0'. Not used when Disable = 1.
Swap[7:0]	Configures the byte swap for the output data. Defines the byte-wise mapping from internal data Pipe[31:0] to output data Out[31:0]. If not set as a one-to-one mapping, undefined behavior or data overlap may occur. Not used when Disable = 1.

Value	Swap[7:6]	Swap[5:4]	Swap[3:2]	Swap[1:0]
	Out[31:24]	Out[23:16]	Out[15:8]	Out[7:0]
0	Pipe[31:24]	Pipe[23:16]	Pipe[15:8]	Pipe[7:0]
1	Pipe[23:16]	Pipe[15:8]	Pipe[7:0]	Pipe[31:24]
2	Pipe[15:8]	Pipe[7:0]	Pipe[31:24]	Pipe[23:16]
3	Pipe[7:0]	Pipe[31:24]	Pipe[23:16]	Pipe[15:8]

SZ	When Stride is 0, use Stride + 1. If set to '0', the value is interpreted as 0x10000; if set to '1', the value is interpreted as 0.
Sw	Specifies the image width used for clipping. If set to '0', the

width is interpreted as Stride + 1; if set to '1', it is interpreted as Delta + 1.

Disable — When set to '1', memory access related to output data is not performed.

Mask — When set to '1', a data mask is applied during memory access for the output data. Not used when Disable = 1.

Exp[1:0] — Configures the detailed pixel format of the output data. Not used when Disable = 1.

Format[1:0] — Sets the Bpp (bits per pixel) of the output data format. For 16Bpp YUYV format, the R component is output to the upper 8 bits, while the G and B components are alternately output to the lower 8 bits depending on whether the horizontal output pixel position is even or odd. Not used when Disable = 1.

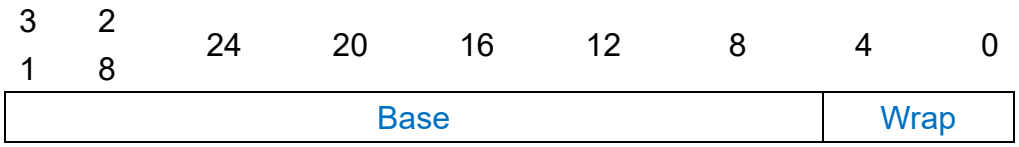
4o

Format	Exp	Out [31:24]	Out [23:16]	Out [15:8]	Out [7:0]	Note
0 8Bpp	0	Pipe[7:0]				-
	1-3	Reserved				
1 16Bpp	0	Unknown		Pipe [23:19] [15:10] [7:3]		RGB565 Lower Cut
	1-3			Pipe[15:0]		-
2 24Bpp	0-3	Unknown	Pipe[23:0]			-
3 32Bpp	0-3	Pipe[31:0]				-

Out: Memory side
Pipe: Blender side (=ARGB)

5.3.1.8. DstBase Command

[Address: 0x1c]



Name	Description
Base[31:6]	Sets the base address of the output data. Must be specified in 64-byte alignment units (see also DstInfo). Not used when DstInfo.Disable = 1.
Wrap[5:0]	See DstBase. Not used when DstInfo.Disable = 1.

6. Application Note

6.1. Overall Control

6.1.1. On Processing Units

- The frBLT converts a chunk of data from the intermediate (parameter) coordinate system into real-world coordinates before executing processing. The shorter the chunk size, the more quickly it can switch between different tasks; however, because access to external memory becomes less contiguous, wasted cycles may occur when switching accesses.
- Loading a command list takes at least 32 cycles. Therefore, for an frBLT capable of processing 1 pixel per cycle, a chunk size of at least 32 pixels is desirable. That said, if the same command list is used repeatedly, the load can be skipped—so unless different command lists alternate very frequently, having a chunk size smaller than 32 pixels poses no practical problem.
- When using pss, you must account for its minimum task-switch latency (2 cycles × the number of pipelines in use). In most cases, simply setting the task size equal to the total number of horizontal pixels in the image is sufficient.

7. Revision History

Version	Changes	Date
1.5	– Added “3.12 Constraints” – Unified terminology (e.g. “out-of-region,” “four-dimensional”)	2021/11/02