

Jasmine Memory Calculations

MB87P2020-A

© Fujitsu Microelectronics Europe GmbH

History

Date	Author	Version	Comment
17/10/03	MMu	V1.0	First version

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products** (eg. software include or header files, application examples, application Notes, target boards, evaluation boards, engineering samples of IC's etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. **Note, all these products are intended and must only be used in an evaluation laboratory environment.**

1. Fujitsu Microelectronics Europe GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days from the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Microelectronics Europe GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.
2. Should a Product turn out to be defect, Fujitsu Microelectronics Europe GmbH's entire liability and the customer's exclusive remedy shall be, at Fujitsu Microelectronics Europe GmbH's sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Microelectronics Europe GmbH in original packing and without further defects resulting from the customer's use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.
3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.
4. To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH's and its suppliers' liability is restricted to intention and gross negligence.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

To the maximum extent permitted by applicable law, in no event shall Fujitsu Mikroelektronik GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect.

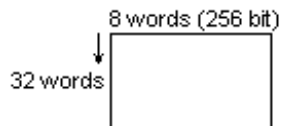
Jasmine memory calculation for layers

The physical address offset describes the start address of a layers position.

Due to the block structure of the memory data, the part of the row address is valid for the physical start address offset only (bits [19:10]).

Domain size in X-dimension DSZ_X is given in logical pixels. It is needed to calculate the pixels per line. The Y-dimension is not needed. Color space code $CSPC_CSC$ is a representation of the appropriate color format. It is converted internally to calculate the bits per pixel (bpp), which is equal to the number of bits the pixel address has to be shifted to get the right word address.

Layer memory can only be divided into whole numbered parts of rows in each dimension. Each block segment is defined to 8 words horizontal (number of pixels depending on color depth (bpp)) and 32 lines vertical.



Following examples demonstrate physical address calculation with relationship to color depth (bpp).

1bpp – 320x240

256 pixels in X-Dim (256 bit / 1 bpp)

32 pixels in Y-Dim

$320 / 256 = 1,25 \rightarrow 2$ blocks horizontal $\Rightarrow 2 * 256 = 512$ pixels

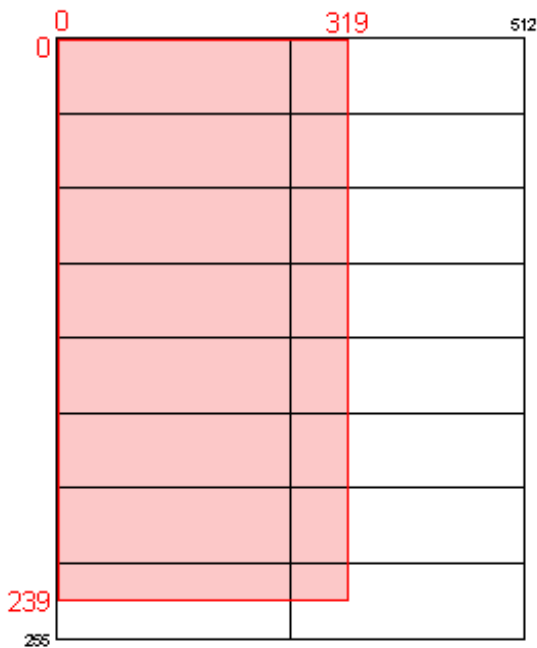
$240 / 32 = 7,5 \rightarrow 8$ blocks vertical $\Rightarrow 8 * 32 = 256$ pixels

layer active size: 320x240

layer physical size: 512x256

physical address offset for the next layer:

$2_{\text{blocks}} * 8_{\text{blocks}} * 8_{\text{words}} * 32_{\text{words}} * 4_{\text{byte}} = 16384$ byte = 16 Kbytes memory for one layer



The offset to the next layer is 0x04000 in the memory:

L0: 0x00000

L1: 0x04000

L2: 0x08000

L3: 0x0C000

L4: 0x10000 ...

2bpp – 320x240:

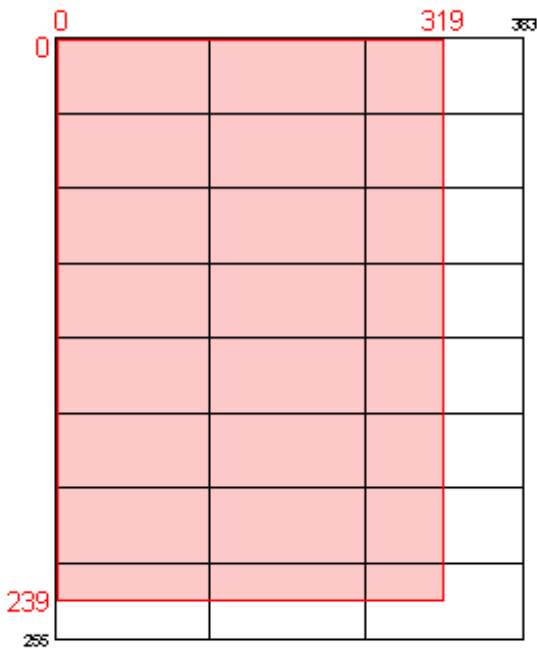
128 pixels in X-Dim (256 bit / 2 bpp)
32 pixels in Y-Dim

$320 / 128 = 2,5 \rightarrow 3$ blocks horizontal $\Rightarrow 3 * 128 = 384$ pixels
 $240 / 32 = 7,5 \rightarrow 8$ blocks vertical $\Rightarrow 8 * 32 = 256$ pixels

layer active size: 320x240
layer physical size: 384x256

physical address offset for the next layer:

$3_{\text{blocks}} * 8_{\text{blocks}} * 8_{\text{words}} * 32_{\text{words}} * 4_{\text{byte}} = 24576$ byte = 24 Kbytes memory for one layer



The offset to the next layer is 0x06000 in the memory:

- L0: 0x00000
- L1: 0x06000
- L2: 0x0C000
- L3: 0x12000
- L4: 0x18000 ...

RGB111: 8bpp – 320x240:

32 pixels in X-Dim (256 bit / 8 bpp)
32 pixels in Y-Dim

$320 / 32 = 10 \rightarrow 3$ blocks horizontal $\Rightarrow 10 * 32 = 320$ pixels
 $240 / 32 = 7,5 \rightarrow 8$ blocks vertical $\Rightarrow 8 * 32 = 256$ pixels

layer active size: 320x240
layer physical size: 320x256

physical address offset for the next layer:

$10_{\text{blocks}} * 8_{\text{blocks}} * 8_{\text{words}} * 32_{\text{words}} * 4_{\text{byte}} = 81920$ byte = 80 Kbytes memory for one layer

The offset to the next layer is 0x14000 in the memory:

L0: 0x00000
L1: 0x14000
L2: 0x28000
L3: 0x3C000
L4: 0x50000 ...

RGB565: 16bpp – 640x480 (e.g. for video input)

16 pixels in X-Dim (256 bit / 16 bpp)
32 pixels in Y-Dim

$640 / 16 = 40 \rightarrow 40$ blocks horizontal $\Rightarrow 40 * 16 = 640$ pixels
 $480 / 32 = 15 \rightarrow 15$ blocks vertical $\Rightarrow 15 * 32 = 480$ pixels

layer active size: 640x480
layer physical size: 640x480

physical address offset for the next layer:

$40_{\text{blocks}} * 15_{\text{blocks}} * 8_{\text{words}} * 32_{\text{words}} * 4_{\text{byte}} = 614400$ byte = 600 Kbytes memory for one layer

The offset to the next layer is 0x96000 in the memory:

L0: 0x000000
L1: 0x096000
L2: 0x12C000
L3: 0x1C2000
L4: 0x258000 ...

The remaining memory of a block cannot be used for the next layer.