

**FUJITSU SEMICONDUCTOR
HARDWARE MANUAL**

GDC

Graphics Display Controller

DMA Application Note



Revision History

Version	Date	Remark
1.1	07. Feb 2001	Initial Release
1.2	19. Mar 2001	Problem solution reviewed

File: *dma_appl_note.fm*

Copyright © 2001 by European MCU Design Centre (EMDC)
Fujitsu Microelectronics Europe GmbH
Am Siebenstein 6-10
63303 Dreieich-Buchsschlag, Germany

This document contains information considered proprietary by the publisher. No part of this document may be copied, or reproduced in any form or by any means, or transferred to any third party without the prior written consent of the publisher. The document is subject to change without prior notice.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (e.g. software include or header files, application examples, application Notes, target boards, evaluation boards, engineering samples of ICs 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 Microelectronics Europe 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.

Table of Contents

	Revision History	2
	Warranty and Disclaimer	3
	Table of Contents	4
1	DMA overview	5
1.1	DMA signals	5
1.2	DMA modes and protocol	5
1.2.1	Overview	5
1.2.2	Level triggered DMA (demand mode)	5
1.2.3	Edge triggered DMA (block-, step-, burstmode)	6
1.3	DMA signal sampling	6
1.4	System behaviour for additional transfers	9
1.5	Board delay calculation	9
2	Problem solutions	11
2.1	Problem solutions for GDC (Lavender, MB87J2120)	11
2.2	Problem solutions for GDC-DRAM (Jasmine, MB87P2020)	11

1 DMA overview

1.1 DMA signals

The MCU DMA-Controller needs beside the normal bus signals¹ for data transfer two additional signals: DMA request (DREQ) from GDC to MCU and DMA acknowledge (DACK) from MCU to GDC. Figure 1-1 shows a block diagram with the required signals.

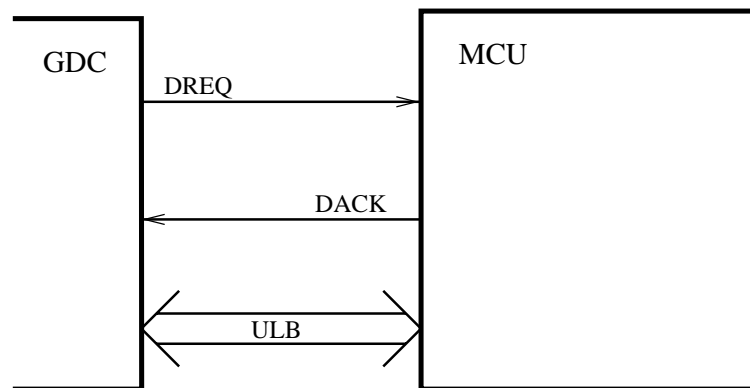


Figure 1-1: DMA connection between GDC and MCU

The protocol is organized as a two way handshake. The external device gives a request (DREQ) to MCU and gets an acknowledge (DACK) back if the request is accepted.

1.2 DMA modes and protocol

1.2.1 Overview

The MCU DMA-Controller can deliver/get data in two different ways:

1. DREQ Level triggered (Demand mode)
2. DREQ Edge triggered (Block-, Step- and Burstmode)

For a detailed description of supported DMA modes see MB91360 series hardware manual.

1.2.2 Level triggered DMA (demand mode)

In case 1. the length of the DREQ signal defines the amount of data to be transferred.

An external device (GDC) has to control the length of DREQ impulses according to internal buffer sizes. It is responsible for the division of data stream while the MCU is only controlling the total amount of data to be transferred. In the special case of GDC does this mean that the GDC DMA-Controller counts the amount of free words for input FIFO (write DMA) or the number of words in output FIFO (read DMA). Table 1-1 gives an overview on transfer sizes in different modes for GDC.

Before starting a demand transfer the GDC DMA-Controller tests for DMA start condition², detects the number of words to be transferred, loads a counter with this value and counts this counter to zero. During counting DREQ is set to active. This procedure is repeated until DMA in GDC is disabled.

The GDC DMA-Controller does **not** know the total amount of words to be transferred. It only tries to fill (write DMA) or to flush (read DMA) its FIFOs. At the end of a complete DMA transfer DREQ could still

1. Signals: CSX, RDX, WRX[3:0], A[20:0], D[31:0], RDY

2. For write DMA: IFDMA:LL >= input FIFO load; for read DMA: OFDMA:UL <= output FIFO load.

be active because from GDC's point of view DMA is enabled and input FIFO needs data or output FIFO has to deliver data. After disabling DMA for GDC the DREQ signal goes inactive.

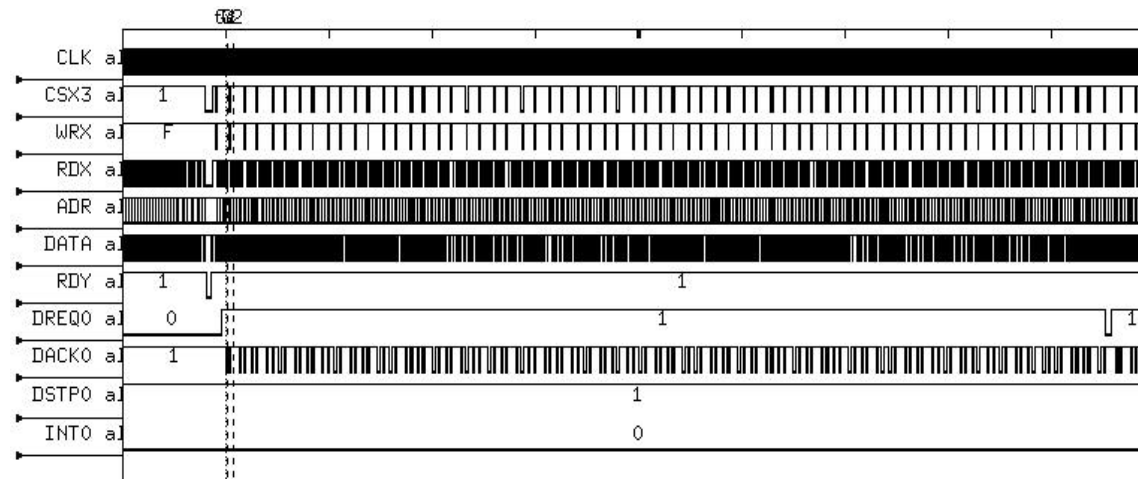


Figure 1-2: Write DMA in demand mode

Figure 1-2 shows the start of a write DMA demand transfer. GDC requests one input FIFO fill cycle¹. During this time a GDC command is active and reads data from input FIFO concurrently. After a short break the second fill cycle is requested.

1.2.3 Edge triggered DMA (block-, step-, burstmode)

In case 2, only the rising edge of DREQ signal is important. The amount of data to be transferred is set within MCU (see also table 1-1).

A MCU peripheral device has to ensure that the DREQ impulse is long enough to be recognized by MCU. In case of GDC the DREQ signal goes inactive after the MCU has acknowledged the DMA request². Depending on MCU mode (block-, step- or burstmode) a MCU defined amount of data words is transferred to or from GDC FIFOs. The programmer has to ensure that no FIFO overflow can occur by setting up the appropriate value for input FIFO lower limit (IFDMA:LL) or output FIFO upper limit (OFDMA:UL).

Table 1-1: Transfer count calculation for DMA

Mode	Input FIFO transfer count [words]	Output FIFO transfer count [words]
Demand mode	trans = FIFO size - FIFO load	trans = FIFO load
Block-, step- or burst mode	trans = <MCU defined>	trans = <MCU defined>

Figure 1-3 shows a write DMA transfer in block mode. The block size is set to 10 words. Despite of this GDC toggles the DREQ signal after every falling edge of DACK signal because GDC does not know the MCU settings. It can not distinguish between block-, step- or burst mode.

1.3 DMA signal sampling

The intention of this chapter is to analyse in which mode and in which data transfer direction (read or write) problems with the delay time for the falling edge of DREQ signal may occur.

1. For Jasmine one complete fill cycle contains 64 words.
2. This is the first high to low edge of the DACK signal combined with a valid chip select signal.

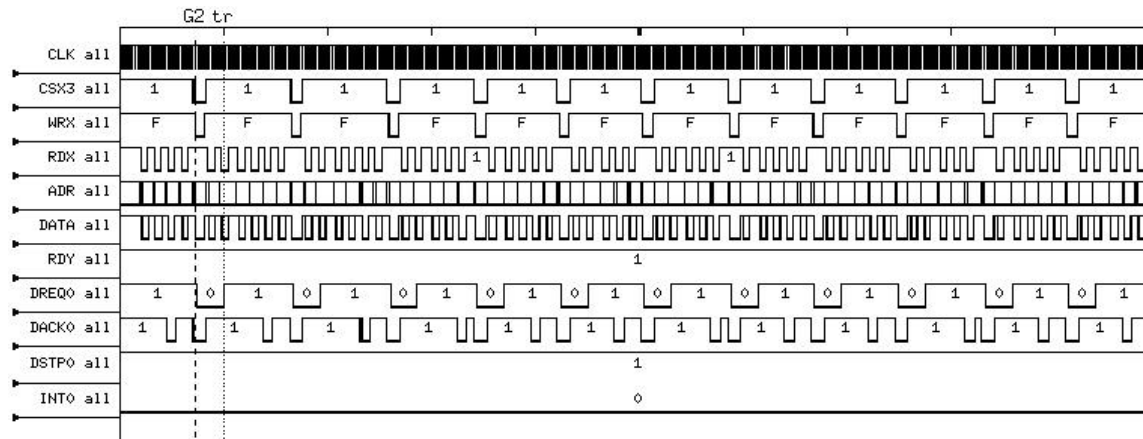


Figure 1-3: Write DMA in block mode

As already mentioned in chapter 1.2 the DMA protocol is different depending on used DMA mode.

Beside these general protocol differences the delay time for falling edge of DREQ signal is different in level (demand mode) and edge triggered (block-, step-, burst mode) DMA transfer.

In edge triggered transfer the DREQ signal goes from active to inactive state (normally the falling edge) one clock after the falling edge of the DACK signal while in level triggered DMA transfer this state change needs three clocks on GDC side.

In level and edge triggered modes additionally the board signal delay for DREQ has to be taken into account. The critical case is a level triggered DMA transfer (demand mode) because the delay is longer than in edge triggered mode. Therefore only demand mode will be taken into account for this document.

Figure 1-4 shows a timing diagram where the system behaviour at the end of a demand transfer is to be seen. It shows the signals for bus interface. The DACK signal is synchronized on MCU bus cycle which means that CSX and DACK switches at the same clock edges. The delay time for falling edge of DREQ signal is measured from the falling edge of DACK signal as already mentioned. Because of the bus cycle synchronization the delay time starts with the same clock edge (see figure 1-4).

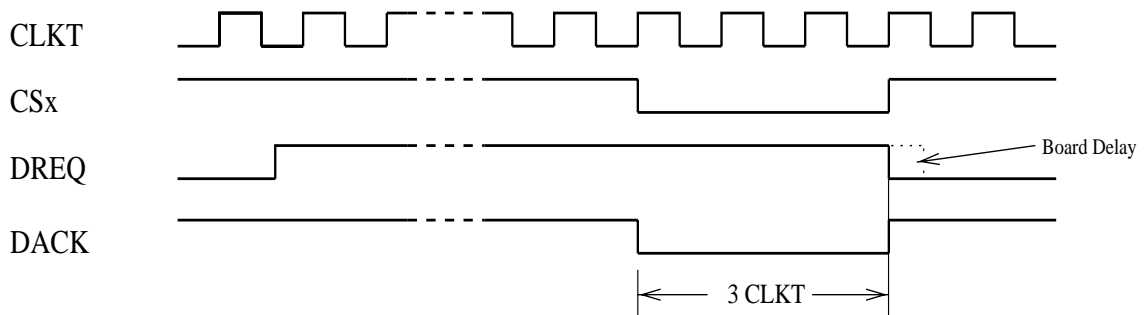


Figure 1-4: Timing diagram for DMA transfer end (demand mode)

The intention of this document was to investigate the maximum board delay for the DREQ signal. An open point is the sample time of MCU. If the MCU sample point for the DREQ signal is too early it may happen that one additional data word is transferred because the MCU DMA-Controller detects the end of DMA access one cycle later (see also figure 1-4). In this cycle the DREQ signal is a long time inactive so it can be ensured that only one (and not more) additional data word is transferred.

In order to find out the DREQ sample time some simulations were performed with GDC-DRAM (Jasmine) and MB91F361 MCU. The software performed a write DMA in demand mode and the parameters were count of waitstates for GDC bus cycles and the delay of DREQ signal¹. This delay models an external (board) delay. As an additional parameter the MCU instruction cache has been turned on and off. It has been observed whether an additional bus access were made or not. Table 1-2 shows the results of these simulations. In figure 1-5 an example for a simulation result is given.

1. In the testbench a generic count of registers can be included in DREQ signal.

Table 1-2: Simulation results for write DMA

Case	Waitstates (Bus cycle length) [clocks]	DREQ delay [clocks]	Instruction cache	Result
1	1 (3)	2	off	no additional cycle
2	1 (3)	3	off	additional cycle
3	1 (3)	4	off	additional cycle
4	1 (3)	0	on	additional cycle
5	1 (3)	2	on	additional cycle
6	1 (3)	3	on	additional cycle
7	2 (4)	3	off	no additional cycle
8	2 (4)	4	off	additional cycle
9	2 (4)	0	on	no additional cycle
10	2 (4)	1	on	additional cycle

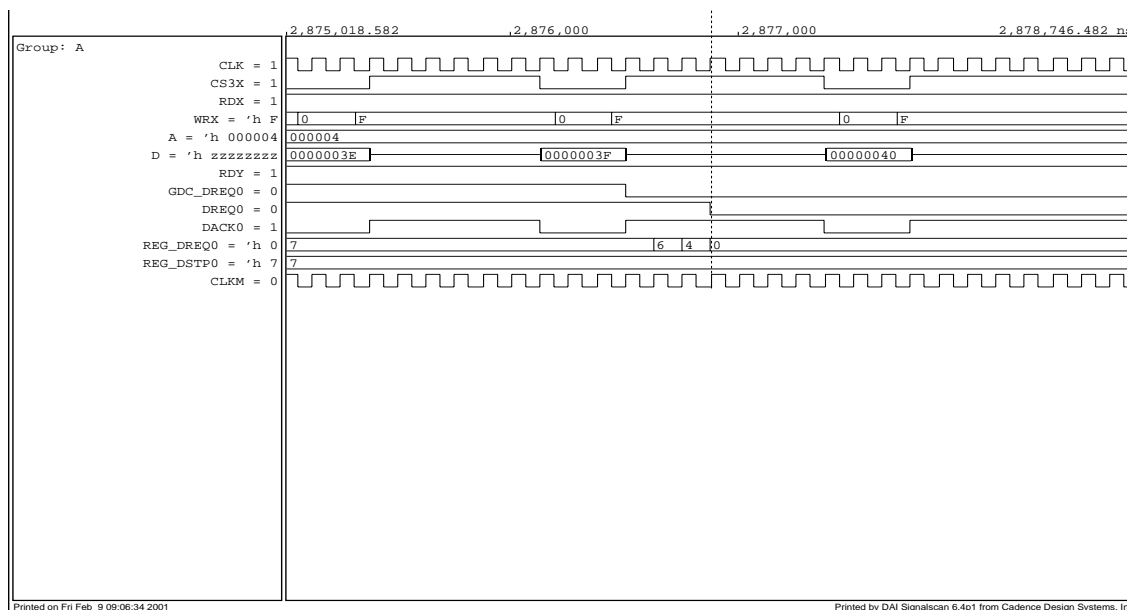


Figure 1-5: Simulation result for case 6 in table 1-2

As to be seen in table 1-2 the MCU behaves different depending whether the instruction cache is turned on or off.

If the cache is turned off the MCU samples the DREQ signal after one additional dummy cycle with exact the length of the 'real' cycle. For instance with one waitstate (bus cycle is 3 clocks long) only delays greater than 3 clocks (case 2 and 3 in table 1-2) causes an additional bus cycle.

If the instruction cache is turned on the MCU samples the DREQ signal at the end of the 'real' bus cycle with the rising edge of CSx. The DREQ signal has to be '0' at this time. As it is to be seen in case 4 in table 1-2 this can not be ensured even with 0 clocks delay. A possible solution is to include one additional waitstate (case 9).

1.4 System behaviour for additional transfers

As the simulations in chapter 1.3 shows under some conditions it is possible that an additional bus cycle is triggered by a delayed falling edge of DREQ. This chapter wants to analyse the system behaviour and the consequences for the application.

Because the total amount of data words is set within MCU (see chapter 1.1) this additional data transfer can **not** increase the total amount of transfer data. It can only cause one additional data word for a FIFO fill or flush cycle. Figure 1-6 shows this situation.

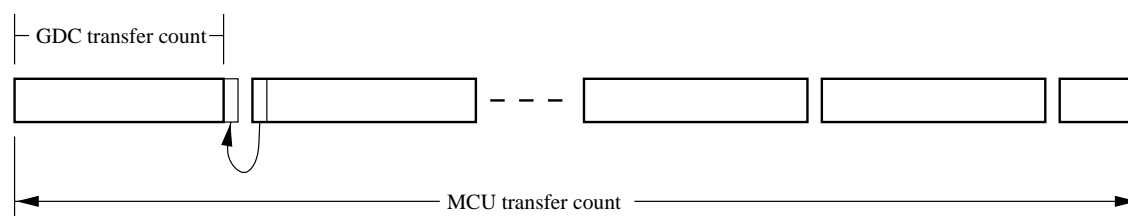


Figure 1-6: DMA transfer division within GDC for demand mode

For write DMA (fill input FIFO) data loss may occur if the GDC transfer count is as large as the input FIFO size and no data are read from input FIFO so that the FIFO is full after FIFO fill process. Only in this case one word from data stream will be lost.

Normally DMA transfer is used to speed up data transfer for large amounts of data (e.g. bitmaps, complex drawings). This means a write DMA transfer is nearly in all cases used together with a GDC command. These commands are waiting for data in input FIFO and start command execution immediately¹. Once the command execution has been started FIFO read access runs concurrently with write DMA transfer. In order to avoid an additional bus write cycle only one data word has to be read during a GDC write transfer cycle (see figure 1-6). The additional delivered data word can be written to input FIFO in this case.

For read DMA (flush output FIFO) wrong data may be transferred if the output FIFO is empty at the end of GDC transfer count (see figure 1-6). This error can **not** occur because for read accesses the RDY signal is used and it can be ensured that the bus cycle is extended by at least two additional cycles. The bus cycle length for reading is then at least five clocks which is more than needed for a save detection of an inactive DREQ signal. In this case the board delay has to be smaller than 2 clocks. Figure 1-7 shows a logic analyser plot with the end of a read DMA cycle in demand mode. In this example a read cycle is 8 clocks long.

1.5 Board delay calculation

In order not to increase DREQ delay over the sample clock edge of MCU the board delay depends on DMA mode and transfer direction (read or write). It is assumed that the instruction cache of MCU is turned on because this the worst case for board delay. With this condition table 1-2 shows that the sample time for the DREQ signal is situated at the end of a bus cycle with the rising edge of CSx. It is also assumed that the set up waitstates are 1 in every case so that a bus cycle takes 3 clocks.

The possible board delay can be calculated with:

$$t_{\text{del}} \leq n \times t_{\text{CLKT}} - t_{\text{OH}_{\text{DREQ}}} - t_{\text{Setup}_{\text{MCU}}}$$

Where t_{CLKT} is one clock period of bus clock (CLKT), $t_{\text{OH}_{\text{DREQ}}}$ is the output hold time of GDC for the signal DREQ and $t_{\text{Setup}_{\text{MCU}}}$ is the setup time of MCU pin; 'n' is the number of possible clock cycles depending on DMA mode and transfer direction. The values for 'n' are collected in table 1-3.

1. Some commands collect a few data words first and start command execution afterwards.

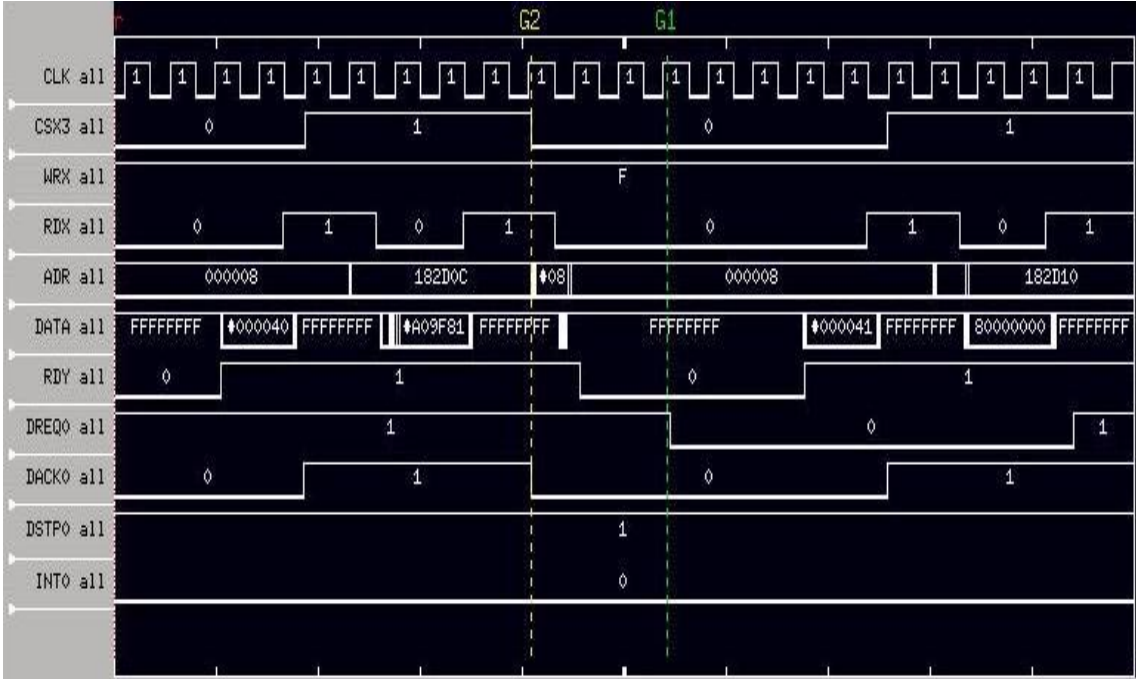


Figure 1-7: Logic analyzer plot for read DMA access

Table 1-3: Maximum board delay cycles ('n' in clocks, one waitstate, instruction cache on)

Mode	Write DMA	Read DMA
Demand mode	-	2
Step mode	2	4
Block mode	2	4
Burst mode	2	4

Only in a write DMA transfer in demand mode under the already mentioned conditions an additional bus cycle is triggered.

2 Problem solutions

2.1 Problem solutions for GDC (Lavender, MB87J2120)

The only possibility to avoid the problem described in chapter 1 is to increase the number of waitstates for Lavender to '2'. The whole bus cycle takes then 4 clocks and the parameter 'n' for delay calculation would become '1'. If this is not suitable and the board delay is higher the count of waitstates can be increased to more than '2'.

2.2 Problem solutions for GDC-DRAM (Jasmine, MB87P2020)

The increment of waitstates according to chapter 2.1 is also a possible workaround for Jasmine but not necessary.

For Jasmine the maximum FIFO size for DMA has been decreased by two words compared to a full FIFO. This means that always two words fewer than the free space in input FIFO are requested for one GDC transfer cycle. The input FIFO in Jasmine can take 66 words but at the most 64 data words are requested from MCU. So it can never happen that a data word is lost even if no FIFO read is performed concurrently (see chapter 1.4).