MPC5200 Local Plus Bus Interface

by Davide Santo and Oliver Bibel

1 Introduction

The Local Plus Bus is the external bus system of the MPC5200 processor, which can be used in conjunction with Flash ROM, SRAM and memory mapped peripherals sharing data bus pins with ATA and PCI devices. This application note gives a brief explanation how to use this interface.

2 General Features of the Local Plus Bus

The Local Plus Bus has a total of eight (8) independent chip selects, 32 Address / Data lines (AD lines) and nine control lines (LP_R/W_b, LP_ALE_b, LP_TS_b, LP_ACK, BANK_SEL[1:0] and TSIZ[1:0] and LP_OE_b). The same physical pins can be used for different functions depending on the interface type and they might not all available at the same time.

On the same data bus up to five PCI devices (four external with the need for an external Priority Encoder for the $\overline{\text{GNT}}$/$\overline{\text{REQ}}$ pairs plus one internal PCI device), an ATA device and up to eight generic memory mapped peripherals like Flash (on different banks), FPGAs, SRAM, E2PROM,etc. could coexist and each of these devices may need a different kind of transaction (the maximum numbers of device connected is limited by the driving power of the Local Plus interface).

For example, PCI uses dual tenure (first the address, then the data are driven on the bus) while Flash might need address and data lines driven at the same time. Another example is ATA which uses a 16-bit wide bus, while external UART controllers usually come with an
8-bit wide bus.

Other time a specific peripheral implements a subset of a PowerPC 60x-like interface using the LP_TS_b, LP_R/W_b and LP_ACK signals (for instance the EPSON-SEIKO LCD controller SD13806) or they might need a separate Read and Write bar with also an Output Enable (e.g. AMD Flash devices as AM29LV065).

Table of Contents

© Freescale Semiconductor, Inc., 2004. All rights reserved.

Launched by Motorola
freescale
semiconductor

2.1 New Local Plus Bus Features of the MPC5200

Two new operating modes called Large Flash and Most/Graphic have been implemented in the Local Plus interface.

They both can be used to BOOT MPC5200 using normal off-the shelf ROM memory devices without the need for external glue logic. The Large Flash interface supports up to 64 Mbytes (it has 26 address lines A25-A0 and 16 bit data lines D16-D0, where the higher ordered number indicates always the Most Significant bit), while the Most/Graphic interface covers up to 16 Mbytes (24 Address by 32 Data bus width). Both have TSIZ signals and LP_OE_b directly available.

The MPC5200 processor has been designed using the 'same' package (PBGA 272 pins) as its predecessor. Therefore the two newly introduced interfaces share now some pins with the PCI/ATA interface preventing them from being simultaneously used at any moment during MPC5200 operations (not only during BOOT).

The basic rule is that PCI is mutually exclusive with the Large Flash or Most/Graphic (but not with the legacy modes of the Local Plus), while ATA can be used with all the legacy modes plus the Large Flash Interface but not with the MOST graphic one.

At power on the PCI controller is in a reset state to avoid possible contention on the Local Plus Bus thus preventing a correct boot process from happening. The PCI controller is taken out of reset via a control bit (bit 16 of the GPIO Port Configuration Register located at MBAR + 0x0B00).

ATA is a chip select controlled interface, then it will not interfere during Boot. It is important to recall that the MGT5200 ATA controller does NOT provide for a reset line to be connected to the SW_RESET (pin #1 of the standard 40 pin ATA connector); this line must be implemented via a GPIO.

Two new Power On Reset Configuration Pins (large_flash_sel connected to ball K02 and most_graphic_sel connected to ball K01) allow the user to select one of the 2 modes for booting.

Only one of the two new modes can be selected at the time; if both POR Configuration pins are set to a logic '1' the LPC will not boot properly or as expected. If none of them is selected then one of the legacy booting mode, either not-muxed or muxed, is automatically used.

As a second new feature for the MPC5200 processor there is the support for today most common burst flash (see as reference the Intel 28F160F3 or AMD29BL802C or ST Microelectronic M58BW016) to reduce the total booting time. Full Bursting is supported for Read only (no write) by the Local Plus controller on the Large Flash, Most/Graphic operating modes.

If the bursting controlled device supports 'Critical Double Word First (CDWF)' wrapping order, then a complete 'cache line' (32 Words) can be read in at the time. When this is not the case, the Local Plus controller can downgrade an internal XLB request to a 'cache word' size (64 bits per beat).

All of this happens *dynamically* in the LPC hardware: if an 8-byte XLB fetch happens (i.e.instruction fetch), then a 64 bit Burst will occur at the peripheral. If an XLB 32-byte cache line fetch happens, then a cache Line burst will occur at the peripheral. It is not possible to burst using an access less that the port size. Bursting mode (64 bit versus CDWF mode) must programmed via SW before first burst transaction can be performed.

Up to three dead cycles can be inserted in a read (Not in a write access!) to better adapt to different peripheral's hold time requirements. Dead cycles are at boot set to the maximum possible (3) and are programmed via SW. The dead cycles (when set to 0 or 1) can be 'hidden' by arbitration cycles and do not add to them.

The Local Plus in the MPC5200 is also hooked to the internal DMA engine (BestComm) allowing data movement to be generated independently from CPU. The interface has been implemented by using a single FIFO (512 bytes deep), which means that the data movement can only happen in a single direction, TX or RX, at the time (i.e. only Half Duplex communication is possible).

Last the Local Plus being shared among different controllers (PCI, ATA, LPC) is arbitrated by a dedicated arbiter which adds 'Bus Parking' as new feature in the MPC5200. 'Bus Parking' is not available for an external Master (such as an external PCI Initiator device).

The Local Plus Bus interface, being so flexible, can be adjusted to meet all its different operating modes by means of software configuration and in some cases minimal external logic (in the multiplexed mode precisely).

NOTE: Multiplexed versus Non-Multiplexed Mode

The most important concept to understand are the Local Plus' two main modes of operation: multiplexed mode and non-multiplexed mode (**Bursting can happen ONLY in a non-multiplexed mode of the Local Plus**).

2.2 How to connect a peripheral to the Local Plus Bus

MPC5200 supports the PCI bus naming convention on the Local Plus Bus. This means that the byte ordering of the external pins conforms to the little endianess rule typical for PCI.

Having a PowerPC core inside, the MPC5200's internal registers, the internal memory and peripherals' bus (XLB and IPBI bus) adhere instead to the big endian ordering rule (IBM style).

The swap is automatically done in the Local Plus module and it is completely transparent to the user. However, it is possible to choose at boot (by using the Hardware Reset Configuration pin called boot_rom_swap connected to ball J02) whether to apply byte swapping or not in case the booting image is a big endian or little endian image.

2.3 MSBs and LSBs

To connect an external device to the Local Plus Bus a couple of simple rules can be used regardless of the mode the bus will use to control the device itself:

- A lower bit number in the pin name indicates lower importance of the line in terms of the bit ordering ('0' is always the least significant bit).
- MPC5200's Local Plus Bus pins are called AD[31:0] where bit 31 is the most significant bit.
- PCI is always 32-bit wide.
- For ATA: Data is connected to AD[15:0] and address to AD[18:16].

2.4 Dynamic Bus Access: The Transfer Size Bits

The Local Plus Bus allows dynamic bus sizing in the terms that on any chip select a transaction of up to 128-bit data width (a Cache line) is allowed and performed transparently to the user.

Dynamic Bus Access refers to any XLB transaction size that is different (larger or shorter) than the LPC data port size, thus forcing the LPC to *dynamically* break the transactions down into multiple LPC bus tenures in order to assemble/disassemble the XLB data transfer onto the LPC bus.

As an example an "instruction fetch" beat (64 bit long) is supported by the Local Plus controller, which translates it into the correct number of external access either 2 for the Most/Graphic interface or 4 for the Large Flash.

The Local Plus controller supports XLB cache-line bursting (32 byte long) assuming that the memory device also support the Critical Double Word First ordering scheme. These modes of the Local Plus are software programmable; refer please to the PowerPC user manual (e.g. refer to "MPC603e & EC603e Risc Microprocessor User Manual") and to the MPC5200 User Manual for further details.

During a Local Plus access using either the legacy multiplexed or the Large Flash or the Most/Graphic mode Transfer Size bits and the address line A0 (LSB) and A1 are put on the bus to reflect the specified transaction similar to the usual dynamic bus access of PowerPC family microprocessors.

Specifically in the Muxed Mode 3 Transfer Size bits TSIZ[0:2] are available (on AD[30:28] during the address tenure), where the most significant one TSIZ[0] is reserved for future use. On the Large Flash and Most graphic mode only TSIZ[1:2] have pinned out. The meaning of the TSIZ[1:2] is in any case the same for the muxed mode and the Large Flash or Most graphic modes.

2.5 Transfer alignment

To illustrate transfer alignments on the bus the following example assumes that the processor performs a transaction where a 32-bit word, e.g. 0x1234ABCD, is read in one 32-bit, two 16-bit or four 8-bit accesses, respectively.

2.5.1 Reading from a 32-bit port

When reading from a 32-bit port *dynamic* bus sizing allows the transfer to take place in a single word access, in two half word accesses or in four consecutive byte accesses. The following tables show the position of the data byte lanes and the value of the TSIZ bits. Please refer to <Blue>Table 11 on page 7 or <Blue>Table 13 on page 18 for the pins where the TSIZ signals will be available depending on the Local Plus mode selected.

Note that in the following tables AD[31:24] is the most significant byte lane while AD[7:0] is the least significant byte lane and that A0 is the least significant bit of the address.

Table 1. Single 32-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	0	0	0	0x12	0x34	0xAB	0xCD

Table 2. Two 16-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
1	0	0	0	0x12	0x34		
1	0	1	0			0xAB	0XCD

Table 3. Four 8-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	1	0	0	0x12			
0	1	0	1		0x34		
0	1	1	0			0xAB	
0	1	1	1				0xCD

2.5.2 Reading from a 16-bit port

If reading from a 16-bit port, then either a 16-bit access or two 8-bit accesses are allowed. To configure the port size the Configuration Register of the used chip select must be written. Note that in a 16-bit chip select the valid data byte lanes are the upper two ones, i.e. AD[31:24] (MSB) and AD[23:16] (LSB).

Table 4. Single 16-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
1	0	0	0	0x12	0x34		
1	0	1	0	0xAB	0XCD		

Table 5. Two 8-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	1	0	0	0x12			
0	1	0	1		0x34		
0	1	1	0	0xAB			
0	1	1	1		0xCD		

2.5.3 Reading from a 8-bit port

With data contents as before, now the chip select is configured as a single byte located on AD[31:24]. Therefore, only four consecutive accesses are possible. It must be noted that this is transparent to the user.

Table 6. Four 8-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	1	0	0	0x12			
0	1	0	1	0x34			
0	1	1	0	0xAB			
0	1	1	1	0xCD			

2.5.4 Write Transaction

This example details a 32-bit write transaction. As before the value 0x1234ABCD is used to write into MPC5200 in one 32-bit, two 16-bit or in four 8-bits accesses.

Table 7. Single 32-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	0	0	0	0x12	0x34	0xAB	0xCD

Table 8. Two 16-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
1	0	0	0	0x12	0x34		
1	0	1	0	0xAB	0XCD	0xAB	0XCD

Table 9. Four 8-bit access

TSIZ1	TSIZ2	A1	A0	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
0	1	0	0	0x12			
0	1	0	1	0x34	0x34		
0	1	1	0	0xAB		0xAB	
0	1	1	1	0xCD			0xCD

2.6 Boot Chip Select

For booting the MPC5200 uses a special chip select called Chip Select Boot ($\overline{\text{CSBOOT}}$). This signal is physically tied to the same ball as chip select 0 ($\overline{\text{CS0}}$). $\overline{\text{CSBOOT}}$ and $\overline{\text{CS0}}$ also share the configuration register located at offset 0x300 from the module base address (MBAR). The only difference between the two 'virtual' chip selects ($\overline{\text{CSBOOT}}$ and $\overline{\text{CS0}}$) lies in the fact that $\overline{\text{CSBOOT}}$ is enabled after reset (during boot) and $\overline{\text{CS0}}$ must first be enabled by software before it can be used as a $\overline{\text{CS}}$ line.

NOTE: Code Execution

Code can be executed from any chip select ($\overline{\text{CSBOOT}}$, $\overline{\text{CS0}}$ through $\overline{\text{CS7}}$).

Also, the two virtual chip select have different mapping registers. $\overline{\text{CS0}}$ start and stop virtual address values are mapped at MBAR+0x04 and MBAR+0x08, while $\overline{\text{CSBOOT}}$ is mapped at MBAR+0x4C and MBAR+0x50.

It is essential to avoid to have both of them enabled (using the enabling control register at MBAR + 0x54) at the same time.

2.7 Local Plus Performance

The Local Plus interface has been originally designed to be an easy-to-use interface not specifically focused on high speed performance. For this reason arbitration of the bus (there can be different masters on the Local Plus as the processor core, an external PCI master and different targets could be accessed such as ATA, PCI or an external memory mapped device) may be performed at every new access. This of course adds some overhead time, which should be taken into consideration when trying to evaluate the speed characteristics and performances of this interface.

As already mentioned 'Bus Parking' has been added but only for internal Master (LP controller, PCI controller and ATA controller). This feature improves performances allowing a Master to have immediate access to the bus without having to re-arbitrate as long as no other requests are pending.

NOTE: Local Plus Bus Reference Clock

MPC5200 has changed the internal reference clock for wait states and transaction state changes from the internal IPBI clock to the PCI clock. As a result all clock counts are with respect to the PCI clock and not the IPBI clock. This was mandated by the fact that the LPC now supports IPBI:PCI clock ratios of 2:1 and 4:1. All transitions have also been synchronized to the rising edge of the PCI clock.

The Local Plus has now a BestComm hook. Therefore Local Plus can directly transfer in a DMA fashion data from the target to BestComm's FIFOs, thus increasing performance. The BestCOMM interface is a Half Duplex only designed on a 512 deep FIFO. Alarm and granularity level can be set via SW and any $\overline{CS}$ can be supported by the internal DMA engine. See <Blue>Chapter 5, <Blue Italic>BestComm Interface to the Local Plus Bus, for more details.

XLB bus processing and arbitration cycles are removed from the Local Plus Controller transaction timing clock counts when the BestComm interface is used.

3 Non-Multiplexed Mode

In non-multiplexed mode, the 32 physical AD pins of the MPC5200 are divided into two separate, non multiplexed groups, precisely the address and data partitions. This allows data and address to be driven at the same time on separate pins.

Eight different configurations of address and data lines are supported.

Table 10. Non-Multiplexed Address/Data Size Options

Mode Name	Address Size	Data Size	TSIZE Bits	BANK Bits	Memory Size [Bytes]	Total Memory Size [MBytes]	Boot Option	Burst Support	ATA Support	PCI Support
LegacyNonMuxed 8	**24**	**8**	**-**	**-**	**16777216**	**16**	**yes**	**no**	**yes**	**yes**
LegacyNonMuxed 8	16	8	-	-	65536	**0.0625**	no	no	yes	yes
LegacyNonMuxed 8	8	8	-	-	256	**0.000244141**	no	no	yes	yes
LegacyNonMuxed 16	**16**	**16**	**-**	**-**	**65536**	**0.0625**	**yes**	**no**	**yes**	**yes**
LegacyNonMuxed 16	8	16	-	-	256	**0.000244141**	no	no	yes	yes
Large Flash 16	**26**	**16**	**2**	**-**	**67108864**	**64**	**yes**	**yes**	**yes**	**no**
Large Flash8	26	8	2	-	67108864	**64**	yes	yes	yes	no
MOST / Graphics 32	**24**	**32**	**2**	**-**	**16777216**	**16**	**yes**	**yes**	**no**	**no**

For example a possible booting configuration uses 24 address and 8 data lines and is thus able to address 2^{24} data bytes, or a total of 16 Mbytes. A second configuration uses 16 address and 16 data lines. Here, 2^{15} address lines point at a half-word each for at a total of 64 kBytes.

NOTE: Byte Addressing

Recall that line A0 can not be used. Any two 'short' addresses [16 bit data] are separated by an 0x02 offset, which clearly indicates that A0 lines is never asserted, or used thus it shall not be connected to the device. The 'A0' pin of the controlled device is usually connected to A1 of the processor. The TSIZ[1:2] pins can sometimes be used in conjunction with A[1:0] to indicate a byte, short or word transfer; in such case a direct connection from A0 (processor) to A0 (device) is possible or necessary.

The Non-Multiplexed Mode is able to interface gluelessly with memory mapped external devices like Flash ROM, E2PROM or SRAM. It is also faster than the multiplexed mode as it provides both data and address in a single tenure.

The following table shows the Local Plus bus signals available in Non-muxed Mode.

Table 11. Local Plus Signals in Non-muxed Mode

Non Muxed Modes (address and data at the same time)

SignalName[MSB:LSB]

	Local Plus Bus Signal Functionality Dedending On Mode Configured					
MPC5200 Signal Name	**LegacyNonMuxed8**	**LegacyNonMuxed16**	**Large Flash16**	**Large Flash16 Read Burst Enabled**	**MOST / Graphics 32**	**MOST / Graphics Read Burst Enabled 32**
LP_R/W_b	Read / Write	Read / Write	Read / Write	Read / Write	Read / Write	Read / Write
LP_ALE_b	-	-	-	-	A23	A23
LP_ACK	Transfer Acknowledge	Transfer Acknowledge	Transfer Acknowledge	Burst Advance	Transfer Acknowledge	Burst Advance
LP_TS_b	Transfer Start	Transfer Start	Transfer Start	Start Address for Burst	Transfer Start	Start Address for Burst
LP_OE_b	Output Enable	Output Enable	Output Enable	Output Enable	Output Enable	Output Enable
CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]
LP_AD[31:24]	D[7:0]	D[15:8]	D[15:8]	D[15:8]	D[31:24]	D[31:24]
LP_AD[23:16]	A[23:16]	D[7:0]	D[7:0]	D[7:0]	D[23:16]	D[23:16]
LP_AD[15:8]	A[15:8]	A[15:8]	A[15:8]	A[15:8]	D[15:8]	D[15:8]
LP_AD[7:0]	A[7:0]	A[7:0]	A[7:0]	A[7:0]	D[7:0]	D[7:0]
TSIZE_1	-	-	TSIZE1	TSIZE1	TSIZE1	TSIZE1
TEST_SEL_1	-	-	TSIZE2	TSIZE2	TSIZE2	TSIZE2
PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK
PCI_PAR	-	-	A16	A16	A0	A0
PCI_C/BE_0	-	-	A17	A17	A1	A1
PCI_C/BE_1	-	-	A18	A18	A2	A2
PCI_C/BE_2	-	-	A19	A19	A3	A3
PCI_C/BE_3	-	-	A20	A20	A4	A4
PCI_TRDY	-	-	A21	A21	A5	A5
PCI_IRDY	-	-	A22	A22	A6	A6
PCI_STOP	-	-	A23	A23	A7	A7
PCI_DEVSEL	-	-	A24	A24	A8	A8
PCI_FRAME	-	-	A25	A25	A9	A9
PCI_SERR	-	-	-	-	A10	A10
PCI_PERR	-	-	-	-	A11	A11
PCI_IDSEL	-	-	-	-	A12	A12
PCI_REQ	-	-	-	-	A13	A13
PCI_GNT	-	-	-	-	A14	A14
PCI_RESET	-	-	-	-	A15	A15
ATA_DRQ	-	-	-	-	A16	A16
ATA_DACK	-	-	-	-	A17	A17
ATA_IOR	-	-	-	-	A18	A18
ATA_IOW	-	-	-	-	A19	A19
ATA_IOCHDRY	-	-	-	-	A20	A20
ATA_INTRQ	-	-	-	-	A21	A21
ATA_ISOLATION	-	-	-	-	A22	A22

3.1 Dedicated LocalPlus Signals

3.1.1 Transfer Acknowledge (LP_ACK)

The use of LP_ACK for termination is software programmable. The LP_ACK, when input, can only reduce the maximum programmed duration of the chip select transaction but **never** prolong it. The same pin can be used as an output in the Most/Graphics and Large Flash interface indicating the *burst advance* of a burst access.

3.1.2 Address Latch Enable (LP_ALE_b)

This signal serves as the additional address line A23 when MOST/Graphics mode is selected.

3.1.3 Transfer Start (LP_TS_b)

This pin can be used as an output in the Most/Graphics and Large Flash mode indicating a valid start address for the next burst transfer.

3.2 Configuration options

3.2.1 Wait States

The numbers for read and write wait states can be set independently. The wait states are counted in PCI clock cycles.

3.2.2 Dead Cycles

Dead cycles after a read access can be inserted (up to 3 as chosen by the user via SW) to provide for proper peripheral hold time. Some dead cycles might be hidden by arbitration.

In either case the data and control signals will be maintained one clock cycle beyond $\overline{CSx}$ negation to assure hold time.

3.3 Booting

At boot by default (via the Hardware Reset Word) the wait states inserted can be either 4 PCI clock cycles or 48 PCI clock cycles. This value can be changed by software to anything from 0 to 127 (or even more when a prescaler is used) once the boot process has successfully started.

Normally a slow device such as Flash ROM memory may need some waits states and therefore at the very boot it might be necessary to start the MPC5200 with the longest wait state default (48 PCI = 727 ns at the maximum allowed Local Plus speed of 66 MHz).

3.4 How to connect a Flash device in a legacy mode

This example assumes the use of a byte wide Flash device with an address space of 8 Mbytes. In this case the 24 address + 8-bit data non-multiplexed mode can be used. The AD pins are split in two groups the following way: AD0-AD22 (which will be renamed A0-A22) and AD24-AD31 (which will be renamed to D0-D7). With this convention the Address and the Data bus can be directly connected to the device. In fact, A0 is indeed the least significant bit of the address section of the Local Plus interface while D0 (connected to AD24 on the processor) is indeed the least significant bit of the data byte lane.

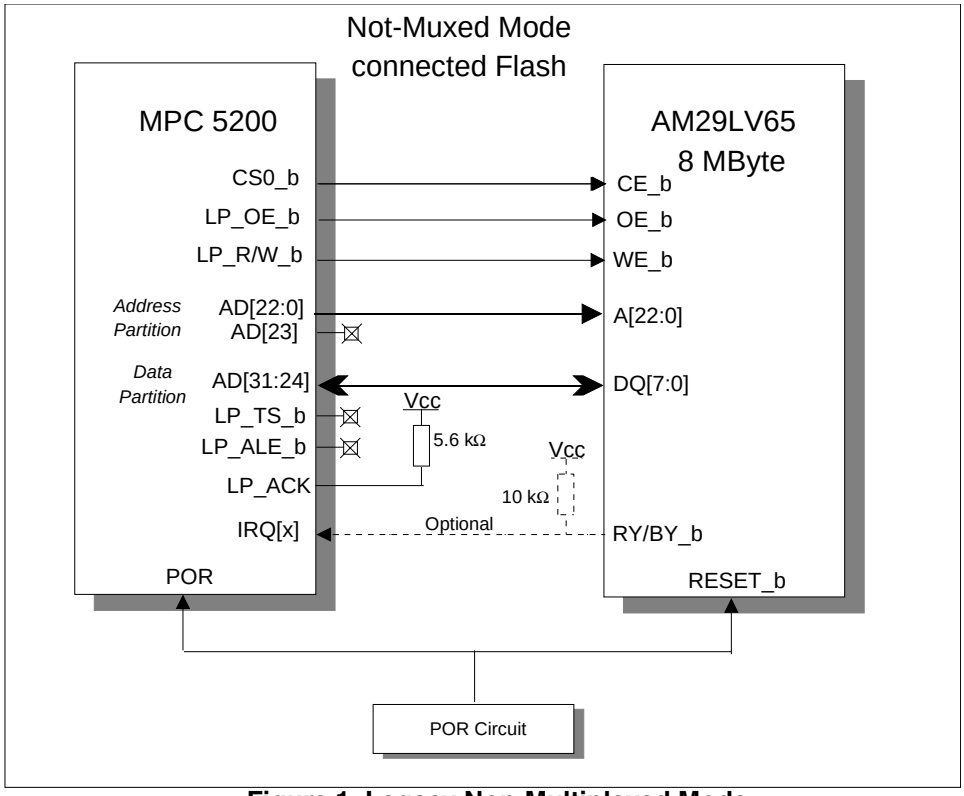


Figure 1. Legacy Non-Multiplexed Mode

3.5 How to connect a Flash device in the Large Flash mode

This example assumes the use of a synchronous 16 Bit wide Flash device with an address space of 1 Mbyte that supports burst reads. Figure 2 shows how the Local Plus pins have to be connected to the flash device. It should be noted that also some PCI signals are used to extend the address range of the Local Plus Bus during Local Plus transfers. A maximum of 26 address lines

(A[25:0]) is available in the Large Flash mode. In this example only A[19:1] are required to cover the whole address range of the flash device connected. A0 is the LSB of the address section of the Local Plus interface and D0 is the LSB of the data byte lane.

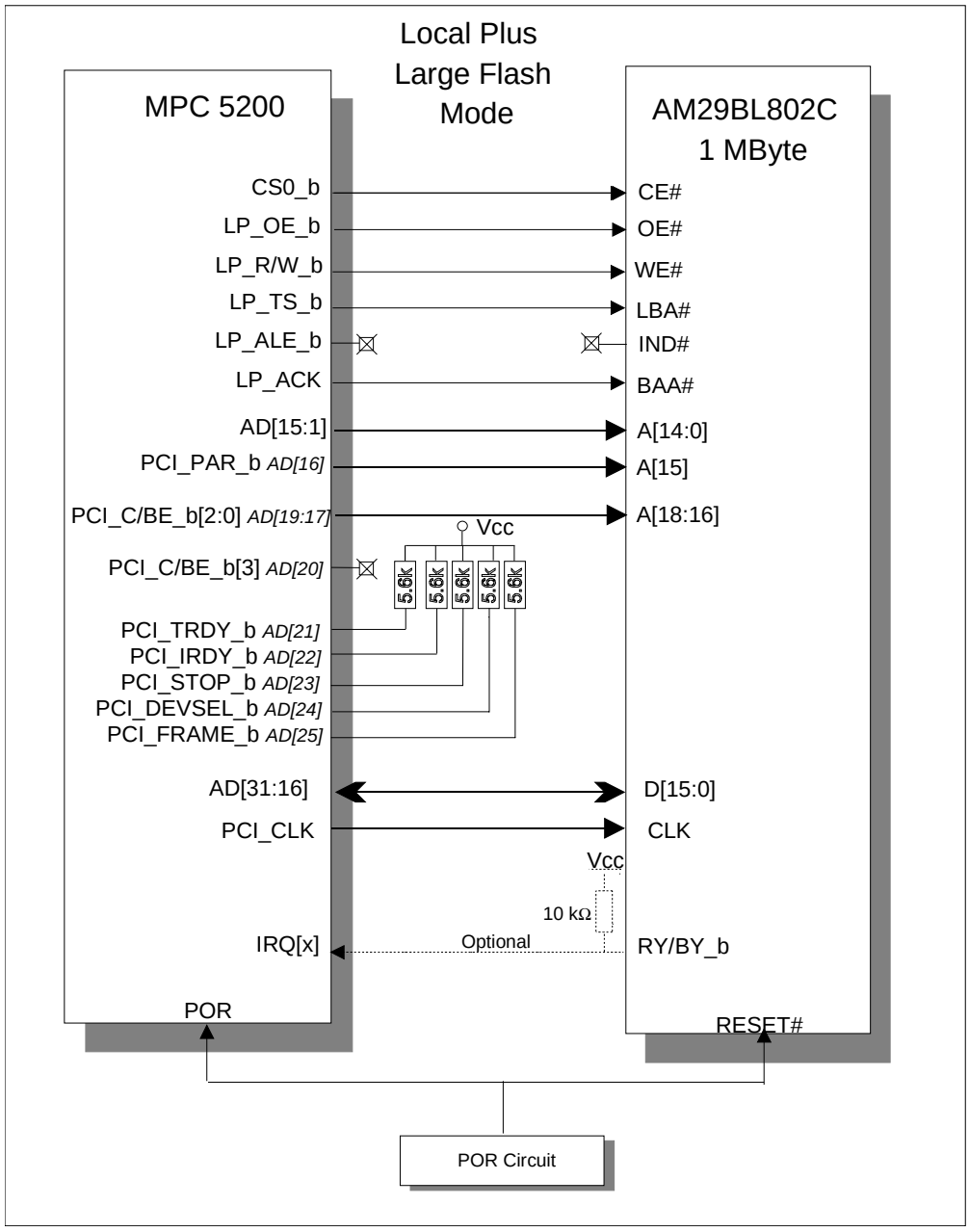


Figure 2. Large Flash Mode

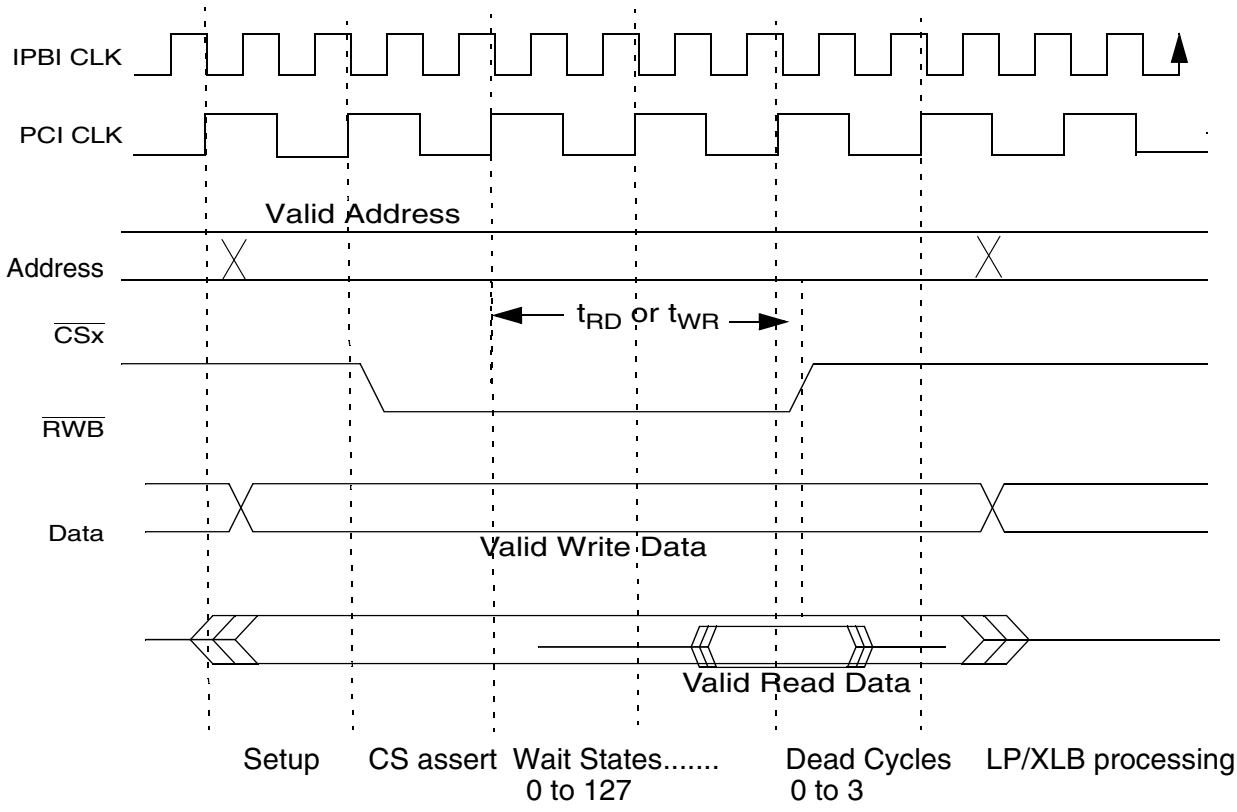
3.6 Timing

Figure 3 shows the timing of transactions on the Local Plus Bus in non-multiplexed mode. It should be noted that the bus states change on the rising edge of the internal PCI clock

To calculate how many clock cycles are involved in a Local Plus transaction the following data will be useful. For detailed values (like Data and Address set up and hold times) please refer to the MPC5200 timing data sheet and User Manual.

The number of clock cycles will be per data beat, which is based on the port size (it could be 8-bit or 16-bit or 32-bit).The reference clock is the PCI clock.

The following figure shows the LocalPlus bus timing in Non-multiplexed modoe.



t_{RD} / t_{WR} are wait states as programmed for corresponding access and chip select.

Read data has nominal setup/hold requirements around the CS negation.

Note that signals are driven with one-clock setup and hold outside of CS active.

Figure 3. Local Plus Non-multiplexed bus timing with 2:1 IPBI to PCI clock ratio

These numbers provide the best case timing, where it is assumed that the PCI arbiter is parked on the Local Plus Controller, there is no other traffic on the internal Bus (XLB), and pipelining of the XLB is enabled. The XLB, for future reference, is the processor core's bus (PPC 60**X** **L**ocal **B**us). Again the Wait states must be computed in terms of PCI clock cycles. XLB pipeline can be turned on/off via SW control (the control register belongs to the XLB Arbiter register group). Pipelining refers to the possibility to start internally on the XLB bus a second address tenure while the data tenure corresponding to the previous address has not yet terminated (either with a TA or a TEA). It might increases marginally performances.

With *Dynamic* Bus Sizing is important to remember that when a wider piece of data is decomposed in smaller chunks to adapt to the peripheral size no additional overhead is added. Nonetheless any time a new data (32 or 64 bit long) is to be read/written from the core a 'turnaround' overhead clock is needed. This is calculated by the formula:

Overhead Clock = 4 x PCI/XLB clock ratio.

Last is to note that while a normal PowerPC data or instruction can be maximum be 32-bit wide, an instruction fetch is always a 64 bit size (to make full usage of the internal Instruction pipeline).

- Write Access
- 1 setup
- 1 CS assertion
- W = number of waits states (0 to 127)
- (4 * pci/xlb clock ratio) *
- 1 LPC processing **
- (4 * pci/xlb clock ratio) ***

* Do not add this cycle if the transaction is a *dynamic* LPC bus tenure.

** Assumes the PCI arbiter is parked on the LPC, else this number is dependent upon Local Plus Bus traffic.

*** Add these cycle(s) to the count if XLB pipelining is turned off.

For an internal clock ratio of the XLB:IPBI:PCI = 4:2:1, a minimum of 4 pci clocks+W per bus tenure is needed for back-to-back writes.

3.6.1 Read Access (non burst)

- 1 setup
- 1 CS assertion
- W = number of waits states (0 to 127)
- D = programmable dead cycles (0 to 3) to allow peripheral time to 3-state bus after read
- 1 LPC processing (this may be masked by dead cycles if set to 1 or greater)
- (4 * pci/xlb clock ratio) *
- 1 LPC processing **
- (4 * pci/xlb clock ratio) ***

* Do not add this cycle if the transaction is a *dynamic* LPC bus tenure.

** Assumes the PCI arbiter is parked on the LPC, else this number is dependent upon Local Plus Bus traffic.

*** Add these cycle(s) to the count if XLB pipelining is turned off.

For an internal clock ratio of the XLB:IPBI:PCI = 4:2:1, a minimum of 5 PCI clocks+W+D per bus tenure is needed for back-to-back reads.

3.7 Waveform Snapshots

The following pages contain snapshots of simulation waveforms for the following Local Plus port configurations.

- Non-muxed 32-bit XLB read transaction to 16-bit address and 16-bit data port, resulting in dynamic Local Plus Controller accesses. Wait States=14, Dead Cycles=0.
- Non-muxed 32-bit XLB write transaction to 16-bit address and 16-bit data port, resulting in dynamic Local Plus Controller accesses. Wait States=4, Dead Cycles=0.
- Non-muxed 16-bit read transaction to 26-bit address and 16-bit data port, Wait States=5, Dead Cycles=0
- Non-muxed 16-bit write transaction to 26-bit address and 16-bit data port, Wait States=5, Dead Cycles=0
- Non-muxed 16-bit short burst read transaction to 26-bit address and 16-bit data port, Wait States=5, Dead Cycles=0

The simulations were run under the following conditions allowing for maximum performance on the Local Plus Bus:

- Clock ratio as indicated in the figures. E.g. clock ratio = 4:2:1 (133MHz, 66MHz, 33MHz)
- No other XLB traffic is present so XLB arbitration is dedicated to Zeppo core read/write access to/from the Local Plus Controller.
- No other Local Plus Bus traffic is present so the PCI arbiter remains parked on the Local Plus Controller.

These simulated conditions are characteristic of the conditions that would exist during boot operations, where pipelined read-read and read-write XLB transactions are not pipelined.

3.7.1 32 bit Read Access to an 16 bit Address and Data Port

Wait State = 14, Dead Cycle = 0

This transaction shows a back-to-back LPC read access. The visible LPC transactions correspond to the first two XLB bus tenures. The second XLB address/data bus tenure is pipelined with respect to the first address/data bus tenure. As a result, XLB arbitration cycles do not impact the LPC as it goes from a write to a read transaction. There is a XLB clock cycle necessary for the XLB data bus tenure to complete and start the next tenure.

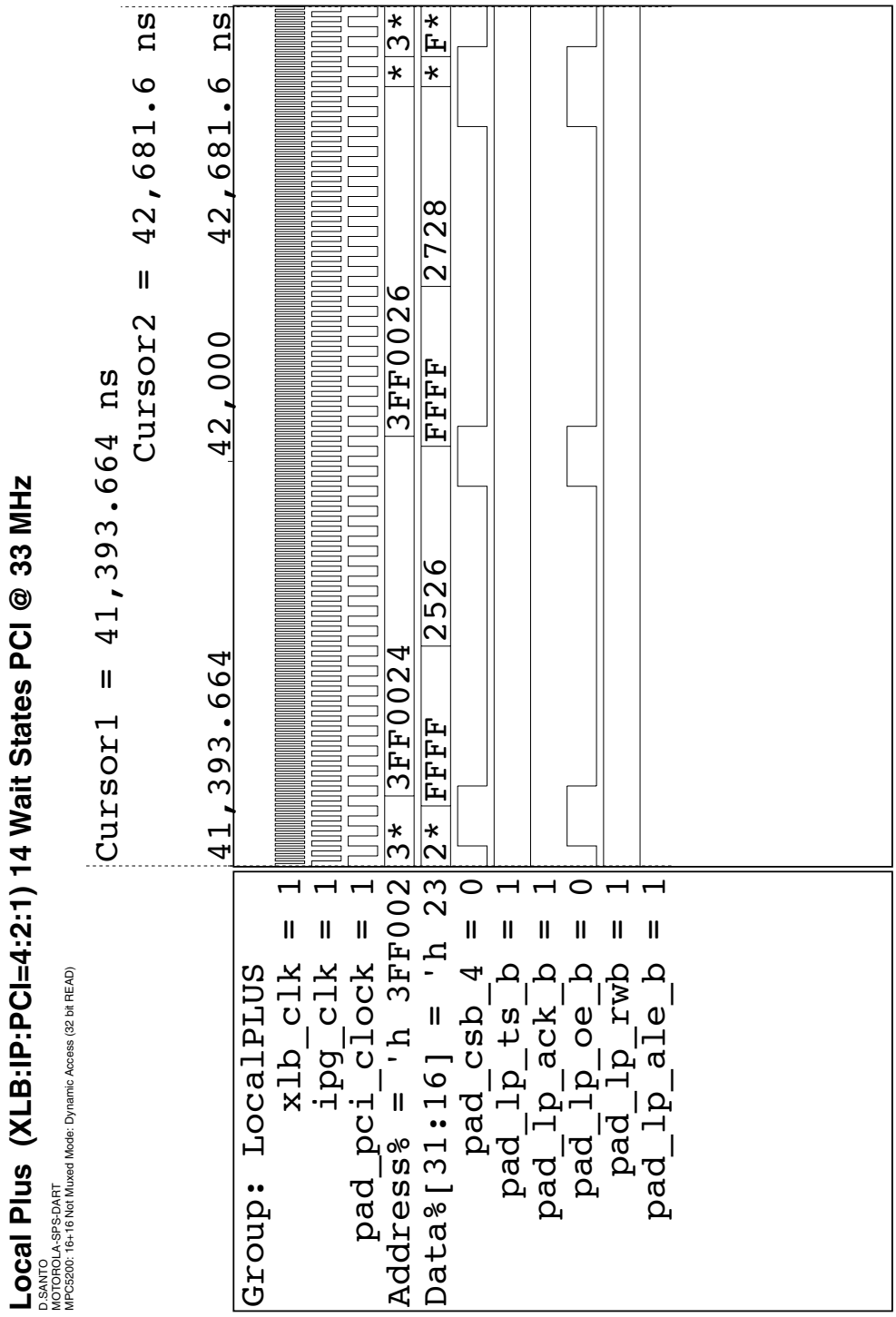


Figure 4. Dynamic Bus Size Read, 14 Wait States, 0 DC

3.7.2 32 bit Write Access to an 16 bit Address and Data Port

Wait States = 14, Dead Cycles = 0.This waveform shows a dynamic bus sizing accesses performed by the LPC. Notice that there is no XLB related PCI clock penalties as the LPC collects the data necessary to complete the XLB data tenure; however, the XLB data tenure now consumes 2 LPC transactions to obtain the necessary data to complete its tenure.

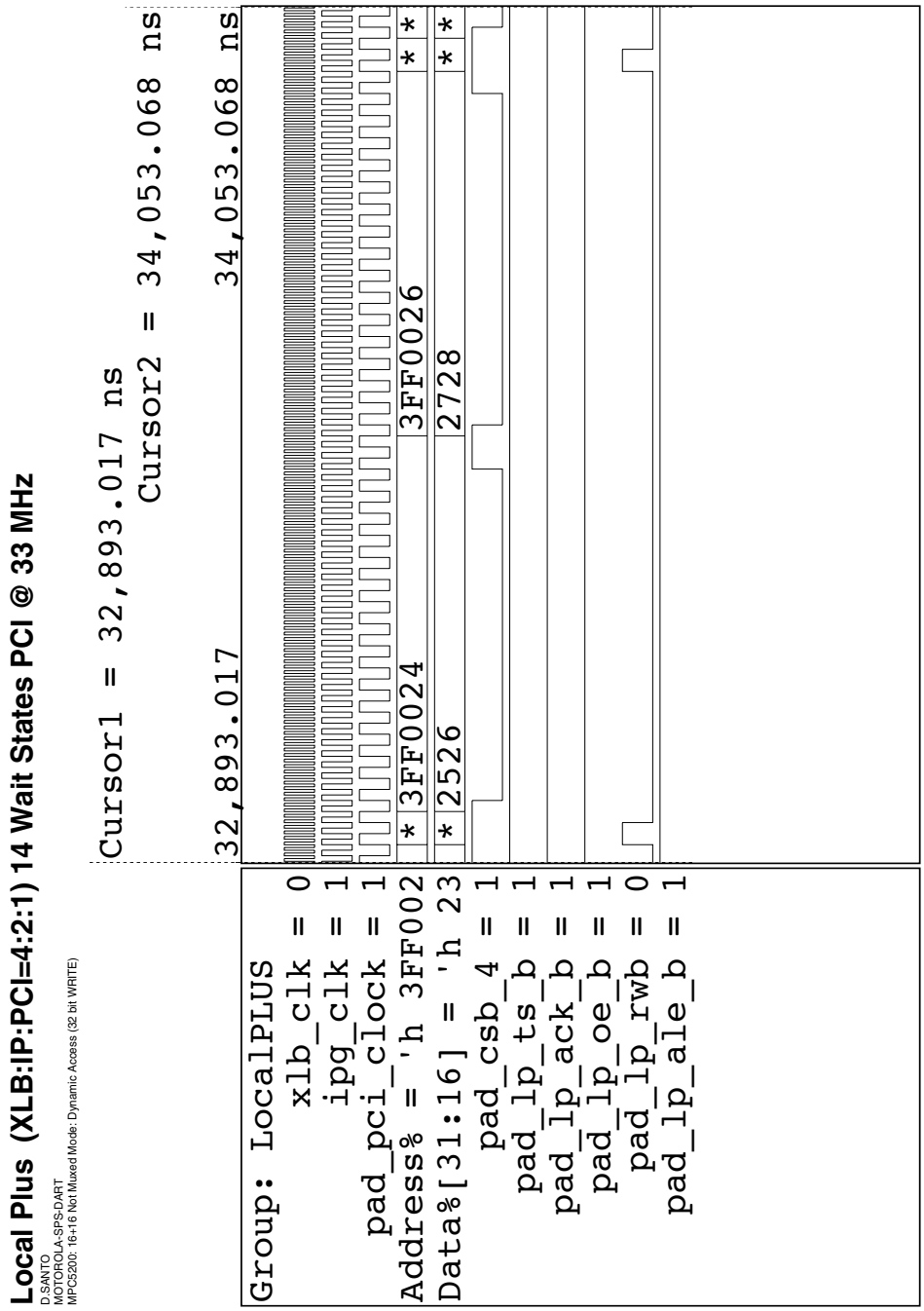


Figure 5. Dynamic Bus Size Write, 14 Wait States, 0 DC

3.7.3 Large Flash Read Access

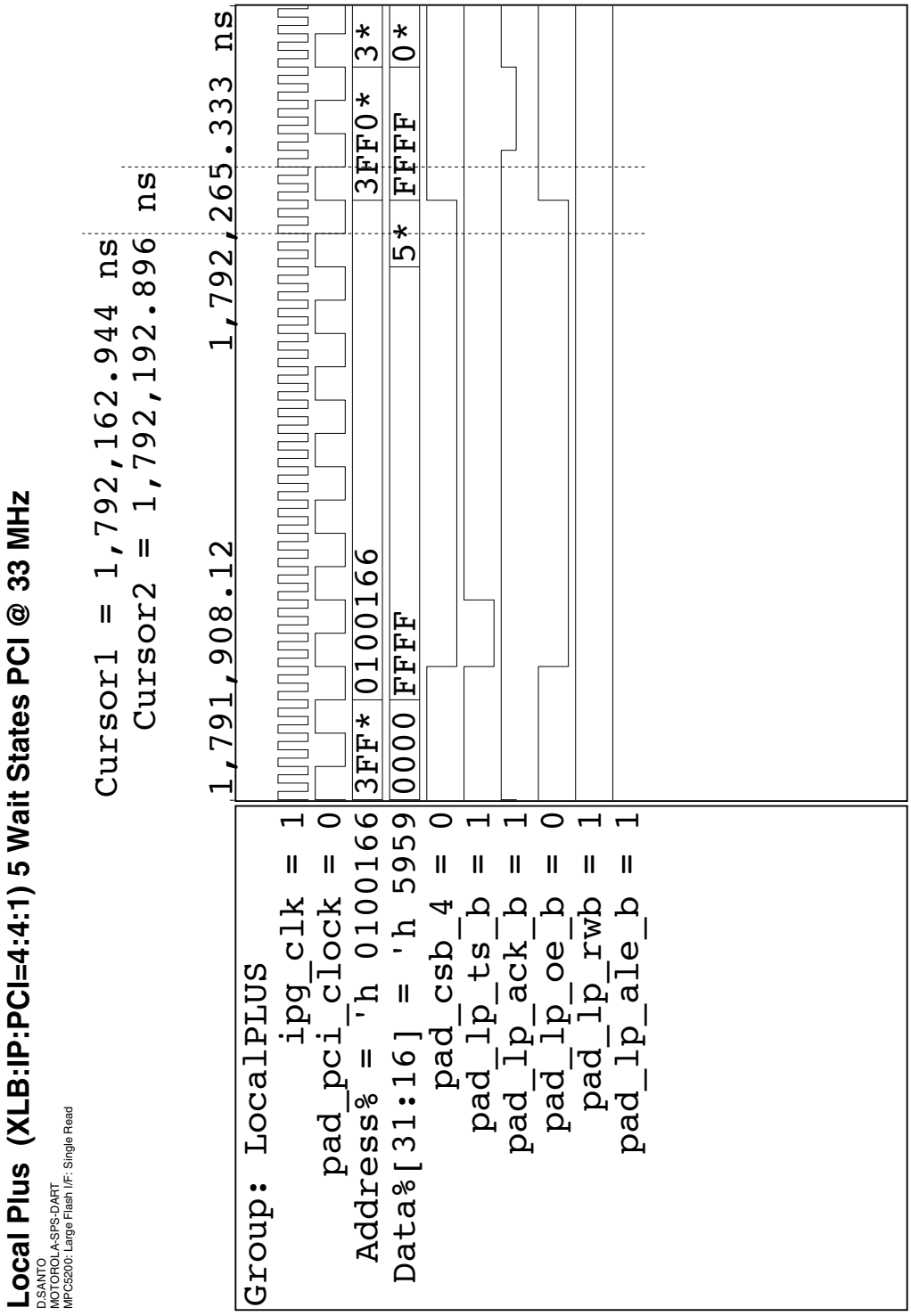


Figure 6. Large Flash Read Access

3.7.4 Large Flash Write Access

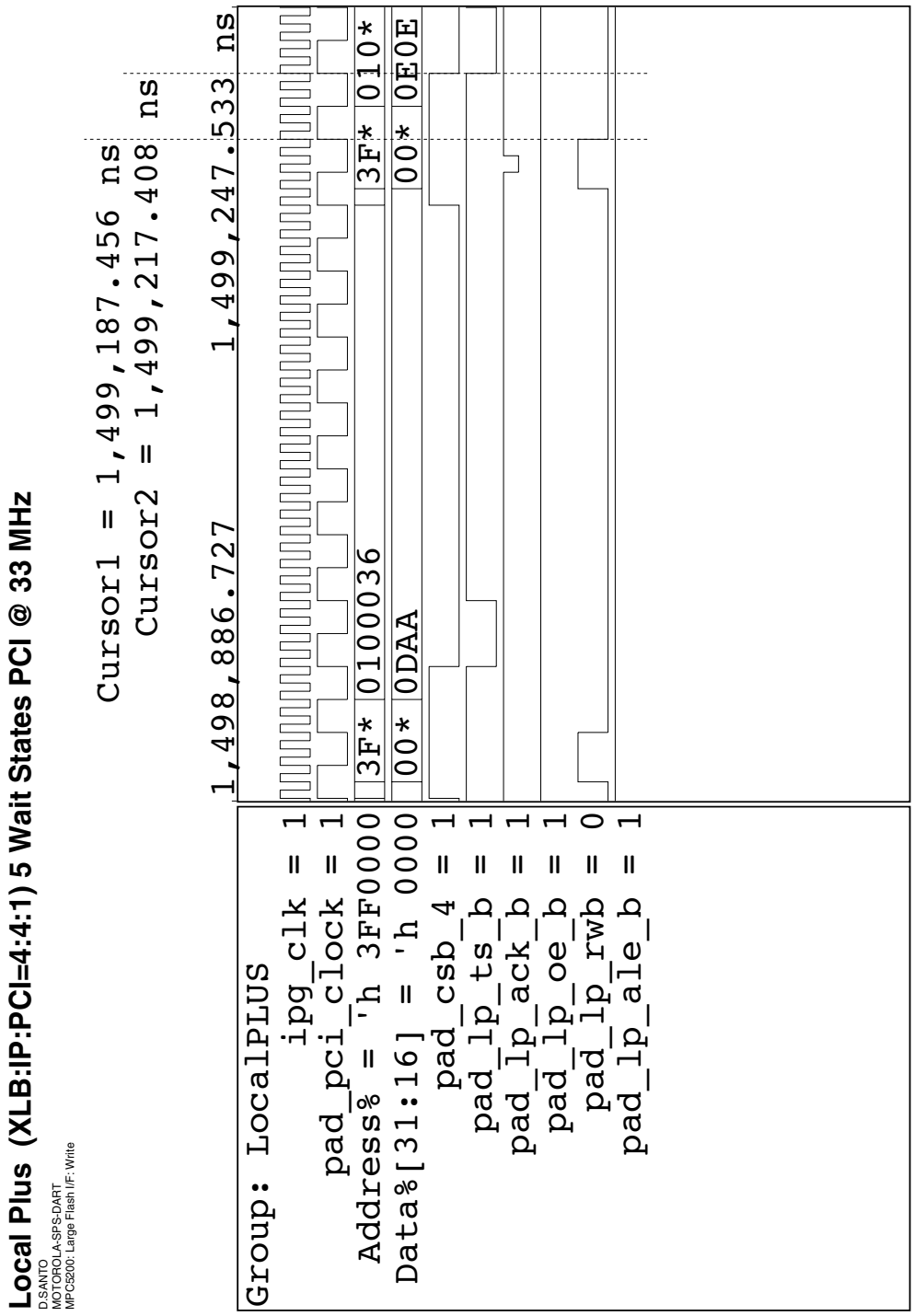


Figure 7. Large Flash Write Access

3.7.5 Large Flash Short Burst Read Access

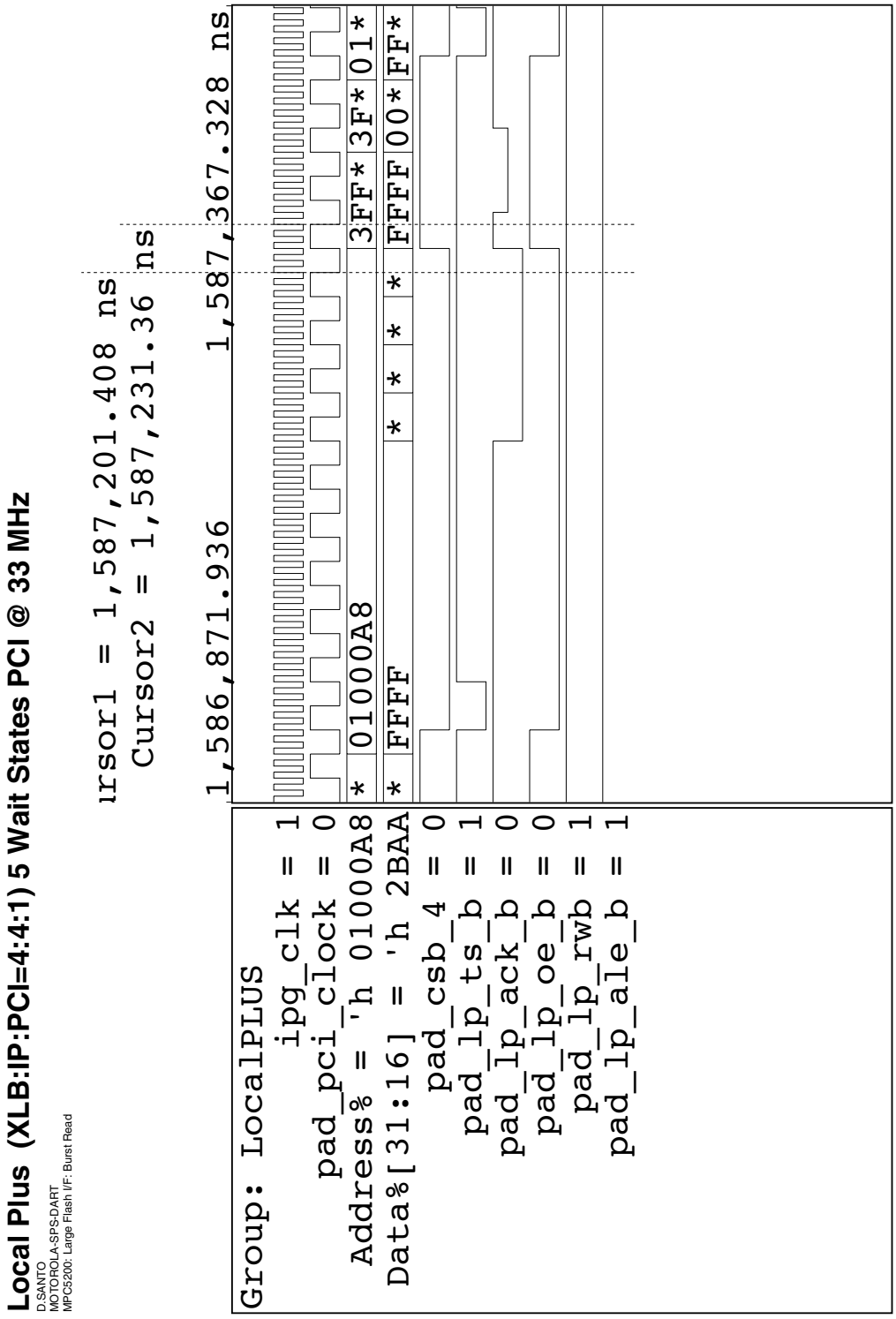


Figure 8. Large Flash Short Burst Read

4 Multiplexed Mode

Opposed to the previous mode, the multiplexed mode implements the address and data lines multiplexed on the same physical pins using dual tenure (or time multiplexing). First, the address is driven on the shared address/data bus and LP_ALE_b is asserted. Then the data is either written or read when the chip select is asserted.

Table 12. Multiplexed Address/Data Size Options

Mode Name	Address Size	Data Size	TSIZE Bits	BANK Bits	Memory Size per Bank [Bytes]	Total Memory Size [MBytes]	Boot Option	Burst Support
LegacyMuxed 8	**25**	**8**	**3**	**2**	**33554432**	**128**	**no**	**no**
LegacyMuxed 8	24	8	3	2	16777216	64	no	no
LegacyMuxed 8	16	8	3	2	65536	0.25	no	no
LegacyMuxed 8	8	8	3	2	256	0.000976563	no	no
LegacyMuxed 16	**25**	**16**	**3**	**2**	**33554432**	**128**	**yes**	**no**
LegacyMuxed 16	24	16	3	2	16777216	64	no	no
LegacyMuxed 16	16	16	3	2	65536	0.25	no	no
LegacyMuxed 16	8	16	3	2	256	0.000976563	no	no
LegacyMuxed 32	**25**	**32**	**3**	**2**	**33554432**	**128**	**yes**	**no**
LegacyMuxed 32	24	32	3	2	16777216	64	no	no
LegacyMuxed 32	16	32	3	2	65536	0.25	no	no
LegacyMuxed 32	8	32	3	2	256	0.000976563	no	no

This approach is slower in comparison to the not-multiplexed mode as it multiplexes the bus functions in time. At least 2 PCI clocks more than in the non-multiplexed case are needed for each bus access. However, the multiplexed mode has the advantage of a bigger address space (up to 128 Mbytes divided in 4 Banks).

It also always needs external logic to latch the address in case this function is not provided by the peripheral. The latch gate must be active low, meaning the latch is transparent when its gating signal is kept asserted to a logic low value.

NOTE: Address Latch Enable

Attention must be paid to the fact that the ALE_b signals deasserts at the same time as the CSx_b will asserts (see <Blue>Figure on page 20). A board trace delay then might cause the LP_ALE_b to be deasserts ‘later’ than the CS falling edge, thus causing Data to be latched instead of an Address. To avoid this problem either a register (driven by the PCI_CLK, with LP_ALE_b gating the clock itself) or an edge-driven latch (using the falling edge) can be used.

The following table shows the Local Plus bus signals available in the Muxed Mode.

Table 13. LocalPlus Signals in Muxed Mode

Muxed Modes (first address tenure followed by the data tenure)

SignalName[MSB:LSB]

Local Plus Bus Signal Functionality Dedending On Mode Configured						
MPC5200 Signal Name	**LegacyMuxed 8**		**LegacyMuxed 16**		**LegacyMuxed 32**	
	Address Phase	**Data Phase**	**Address Phase**	**Data Phase**	**Address Phase**	**Data Phase**
LP_R/W_b	Read / Write	Read / Write	Read / Write	Read / Write	Read / Write	Read / Write
LP_ALE_b	Address Latch Enable	Address Latch Enable	Address Latch Enable	Address Latch Enable	Address Latch Enable	Address Latch Enable
LP_ACK	Transfer Acknowledge	Transfer Acknowledge	Transfer Acknowledge	Transfer Acknowledge	Transfer Acknowledge	Transfer Acknowledge
LP_TS_b	Transfer Start	Transfer Start	Transfer Start	Transfer Start	Transfer Start	Transfer Start
LP_OE_b	Output Enable	Output Enable	Output Enable	Output Enable	Output Enable	Output Enable
CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]	CS[7:0]
LP_AD[31]	0	D7	0	D15	0	D31
LP_AD[30]	TSIZE0	D6	TSIZE0	D14	TSIZE0	D30
LP_AD[29]	TSIZE1	D5	TSIZE1	D13	TSIZE1	D29
LP_AD[28]	TSIZE2	D4	TSIZE2	D12	TSIZE2	D28
LP_AD[27]	0	D3	0	D11	0	D27
LP_AD[26]	Bank Select Bit 1	D2	Bank Select Bit 1	D10	Bank Select Bit 1	D26
LP_AD[25]	Bank Select Bit 0	D1	Bank Select Bit 0	D9	Bank Select Bit 0	D25
LP_AD[24]	A24	D0	A24	D8	A24	D24
LP_AD[23]	A23	0	A23	D7	A23	D23
LP_AD[22]	A22	0	A22	D6	A22	D22
LP_AD[21]	A21	0	A21	D5	A21	D21
LP_AD[20]	A20	0	A20	D4	A20	D20
LP_AD[19]	A19	0	A19	D3	A19	D19
LP_AD[18]	A18	0	A18	D2	A18	D18
LP_AD[17]	A17	0	A17	D1	A17	D17
LP_AD[16]	A16	0	A16	D0	A16	D16
LP_AD[15]	A15	0	A15	0	A15	D15
LP_AD[14]	A14	0	A14	0	A14	D14
LP_AD[13]	A13	0	A13	0	A13	D13
LP_AD[12]	A12	0	A12	0	A12	D12
LP_AD[11]	A11	0	A11	0	A11	D11
LP_AD[10]	A10	0	A10	0	A10	D10
LP_AD[9]	A9	0	A9	0	A9	D9
LP_AD[8]	A8	0	A8	0	A8	D8
LP_AD[7]	A7	0	A7	0	A7	D7
LP_AD[6]	A6	0	A6	0	A6	D6
LP_AD[5]	A5	0	A5	0	A5	D5
LP_AD[4]	A4	0	A4	0	A4	D4
LP_AD[3]	A3	0	A3	0	A3	D3
LP_AD[2]	A2	0	A2	0	A2	D2
LP_AD[1]	A1	0	A1	0	A1	D1
LP_AD[0]	A0	0	A0	0	A0	D0
PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK	PCI_CLK

4.1 Dedicated Local Plus signals

In general the multiplexed mode provides more control signals than the non-multiplexed mode. There is the LP_ALE_b, which is always needed and the LP_TS_b, which is driven low during the first LP clock (the Local Plus clock is called PCI clock), when $\overline{CS}$ has been asserted (to allow peripherals to latch the address internally). The cycle will terminate without an LP_ACK, if the internal wait state condition expires.

Dead cycles after a read access can be inserted (up to 3 as chosen by the User via SW) to provide for proper peripheral hold time.

In either case the data and control signals will be maintained one clock cycle beyond $\overline{CSx}$ negation to assure hold time.

NOTE: Transfer Acknowledge

Use of LP_ACK_b for termination is software programmable. The LP_ACK_b, when input, can only reduce the maximum programmed duration of the chip select transaction, **never** prolong it.

With suitable external glue logic an PowerPC 60x-like bus (the external peripheral bus of some PowerPC family processors like the PPC823e or the PPC8260) with reduced features can be implemented on the MPC5200. What is missing compared to a full *E-bus* are the signal TEA, burst related signaling and the signals to handle multiple masters.

NOTE: Bursting

Bursting is **not** available in any of the Muxed Modes.

4.2 Booting

At boot 8-bit wide multiplexed mode configuration is not allowed. **Only** 16 or 32-bit (32-bit is preferred) are allowed.

Normally a slow device such as Flash ROM memory may need some waits states and therefore at the very boot it might be necessary to start the MPC5200 with the longest wait state default (48 PCI = 727 ns at maximum allowed PCI speed of 66 MHz).

4.3 How to connect an LCD controller in the multiplexed mode

This example shows how a typical LCD controller, the SD 13806 by Epson-Seiko, can be connected to the Local Plus Bus in the Muxed Mode. This peripheral has 21 address lines and 16 separate data lines. It also implements an E-Bus-like interface which makes use of the Transfer Size bits (as the control registers are 8-bit wide while graphic data is 16-bits, so dynamic size access is required), the Transfer Start (LP_TS_b), Acknowledge (LP_ACK) and LP_R/W_b signals.

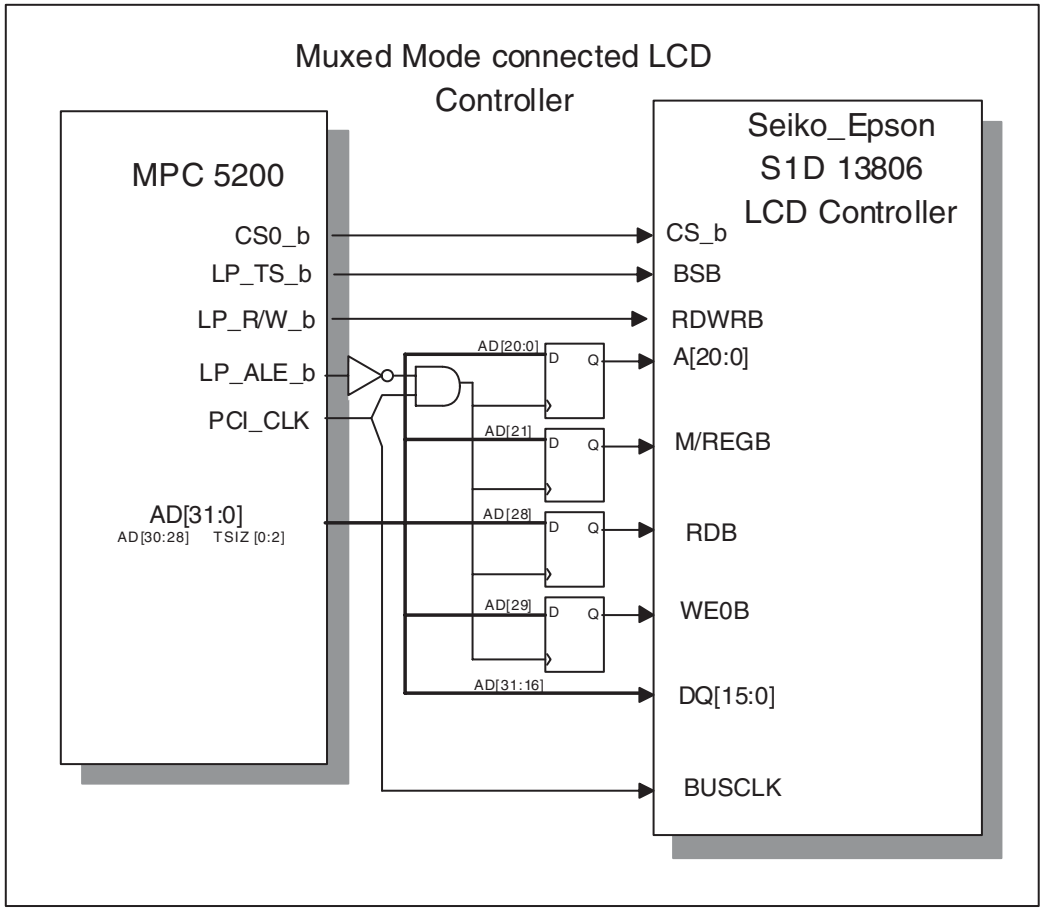


Figure 9. LCD Controller in Local Plus Multiplexed Mode

It is self evident that a legacy non-multiplexed mode is not viable as there would be either not enough address line (16 +16 mode) or not enough data lines (24+8 mode). In this case either an external logic is required to capture the address during the address tenure or the Large Flash interface could be used (which could then interfere with PCI if needed).

The external registers will also store the TSIZ bits (on AD28 the least significant TSIZ bit and on AD29 the most significant) for the legacy mode. For the Large Flash and Most/G interfaces, these bits are directly available from two separated pins.

To connect the device properly the latched address lines (A20-A0) must directly be connected to the A20-A0 pins of the slave device, while for the data part the lines AD31 to AD16 will be connected to the pins named D15 to D0 on the LCD (AD16 again being the least significant bit).

4.4 Timing

Figure 10 shows all general timing relationships of the Local Plus Bus in multiplexed mode. Note that during data tenure, the decision between data being driven or the AD bit being tri-stated is dependent on whether the transaction is a Read or a Write and what the programmed Data Size is. Unused bits in the data tenure (i.e. those in excess of the programmed data size) will be driven to zero by the Local Plus Bus controller as a precaution to avoid floating bus condition.

NOTE: LP_ALE_b

Please note that there is no 'hold time' left between the release of the LP_ALE_b signal an the assertion of the $\overline{CS}$ (or LP_TS_b) signal. This might impact some peripherals and must be carefully taken into consideration.

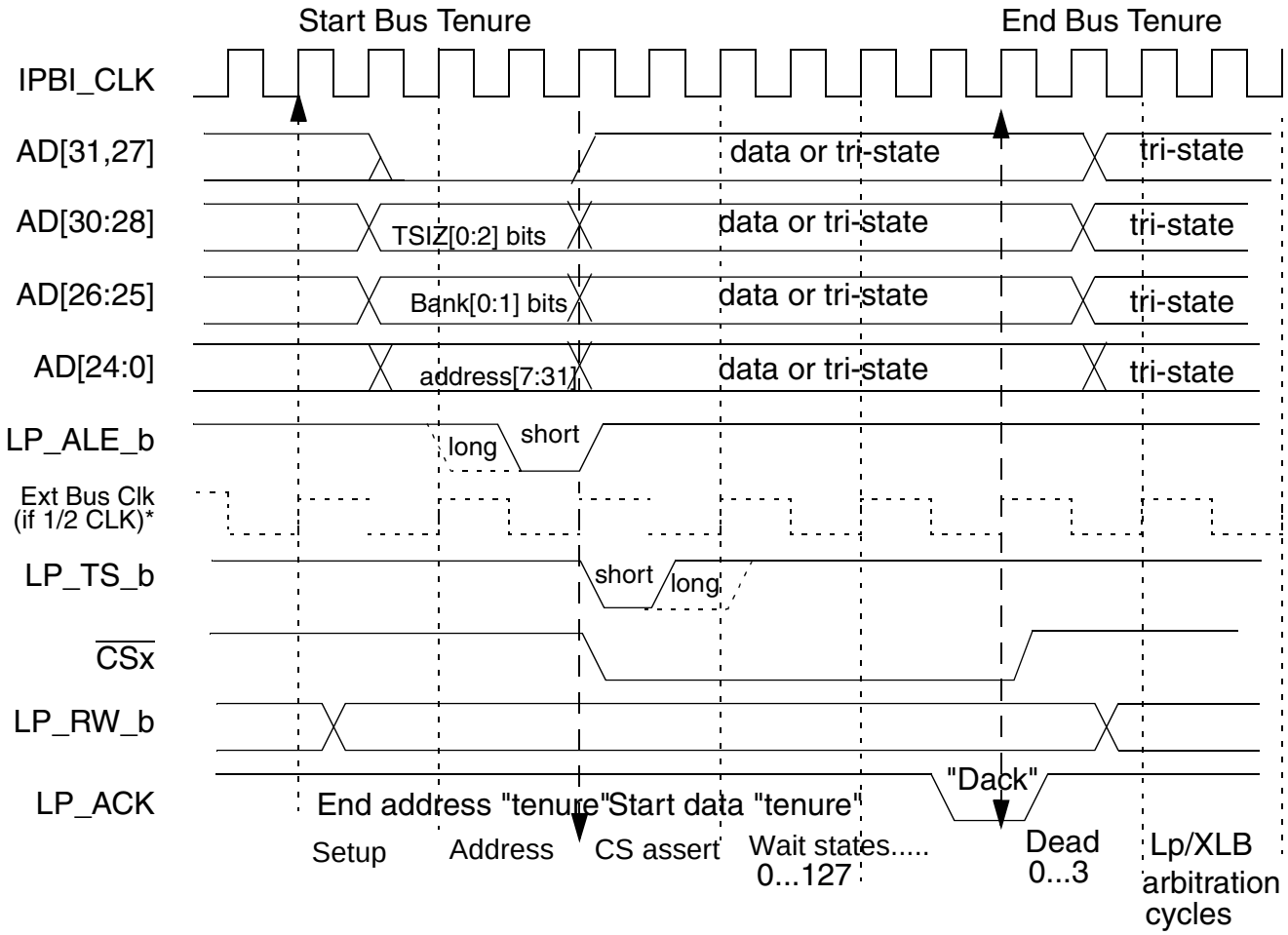


Figure 10. Local Plus multiplexed bus timing.

The maximum address space per each single CS in multiplexed mode is 128 Mbytes divided into four banks each of 32 Mbytes. In fact, there are 25 address line (AD24 to AD0) and two bank select lines (AD26 and AD25) addressing words (32-bits) and that when a 32 bit device is used line A1 and A0 will be NOT connected.

The multiplexed mode also provides the three Transfer Size bits (TSIZ[0:2]). They are latched on AD[30:28] and are used when a dynamic bus access smaller or greater than the nominal bus width is needed.

NOTE: TSIZ0

AD[30] represents TSIZ[0] and it is reserved for future use. This bit is driven to '0' when a 8 bit or 16 bit transaction is performed and to '1' if a 32-bit transaction is executed.

In the Multiplexed Mode the data beat can be either 8, 16 or 32 (only 16 or 32 at boot which is usually the case where this computation affects the most).

4.4.1 Write Access

- 1 setup
- 1 address tenure
- 2 data tenures ($\overline{CS}$ assert)
- W = number of waits states (0 to 127)
- (4 * pci/xlb clock ratio) *
- 1 LPC processing **
- (4 * pci/xlb clock ratio) ***

* Do not add this cycle if the transaction is a *dynamic* LPC bus tenure.

** Assumes the PCI Arbiter is parked on the LPC, else this number is dependent upon Local Plus Bus traffic.

*** Add these cycle(s) to the count if XLB pipelining is turned off.

For an internal clock ratio of the XLB:IPBI:PCI = 4:2:1, a minimum of 6 pci clocks+W per bus tenure is needed for back-to-back writes.

4.4.2 Read Access

- 1 setup
- 1 address tenure
- 2 data tenures ($\overline{CS}$ assert)
- W = number of waits states (0 to 127)
- D = programmable dead cycles (0 to 3) to allow peripheral time to 3-state bus after read
- 1 LPC Processing (this may be masked by dead cycles if set to 1 or greater)
- (4 * pci/xlb clock ratio) *
- 1 LPC processing **
- (4 * pci/xlb clock ratio) ***

* Do not add this cycle if the transaction is a *dynamic* LPC bus tenure.

** Assumes the PCI Arbiter is parked on the LPC, else this number is dependent upon Local Plus Bus traffic.

*** Add these cycle(s) to the count if XLB pipelining is turned off.

For an internal clock ratio of the XLB:IPBI:PCI = 4:2:1, a minimum of 7 pci clocks+W+D per bus tenure is needed for back-to-back reads.

The main difference from the previous mode lies in the fact that an address tenure is needed, where the LP_ALE_b could be kept low at most for one LP (id est PCI) clock cycle (this can be programmed to half a clock by software).

The data tenure minimum time will be 2 PCI clocks.

4.5 External Glue Logic

In case of the Muxed Mode of the Local Plus Bus it is necessary to add external logic to generate control signals needed by a peripheral but not supplied by the Local Plus Bus. As mentioned above an external logic is needed to ‘latch’ or ‘register’ the Address during the Address tenure.

NOTE: LP_OE_b and Write Enable

Not always a pair of signals like LP_OE_b and a Write Enable is needed. Some device simply support a direct connection with the LP_R/W_b.

Some peripherals might also take a particularly long time before releasing the bus after a read access is completed. The Local Plus Bus has a maximum of three dead cycles at the end of a transaction (which can be reduced for a READ only access to 0 via SW programming) where the bus remains tri-stated allowing such peripherals to detach from the bus without the risk of a bus contention.

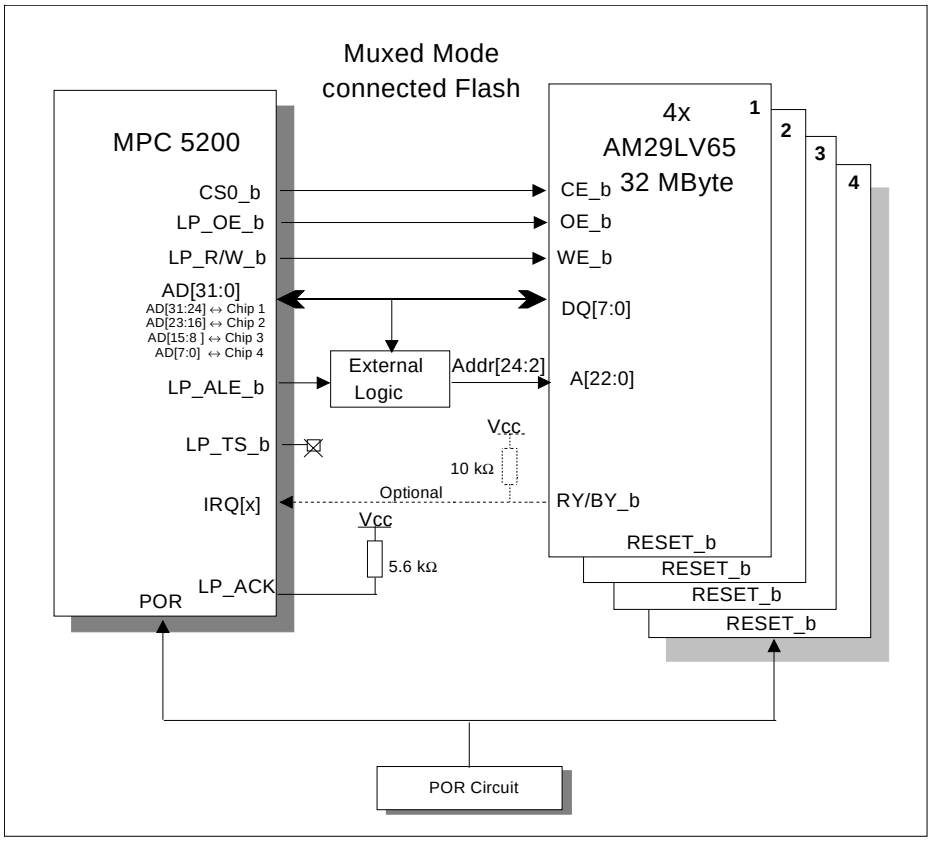


Figure 11. Multiplexed mode address latching

Figure 11 shows how external logic can be used to interface a bank of four 8-bit Flashed devices using the Muxed Mode.

4.6 Waveforms Snapshots

The following pages contain snapshots of simulation waveforms for the following Local Plus port configurations.

- Muxed 32-bit XLB read transaction to 32-bit address and 32-bit data port. ALE=1, Wait States= 8, Dead Cycles= 0.
- Muxed 32-bit XLB write transaction to 32-bit address and 32-bit data port. ALE=1, Wait States= 8, Dead Cycles= 0.
- Muxed 32-bit XLB read transaction to 32-bit address and 32-bit data port. ALE=1, Wait States= 8, Dead Cycles= 2.

The simulations were run under the following conditions allowing for maximum performance on the Local Plus Bus:

- No other XLB traffic is present so XLB arbitration is dedicated to Zeppo core read/write access to/from the Local Plus Controller.
- No other Local Plus Bus traffic is present so the PCI arbiter remains parked on the Local Plus Controller.

These simulated conditions are characteristic of the conditions that could exist during boot operations

4.6.1 Read Access, 8 Wait States, 0 Dead Cycles

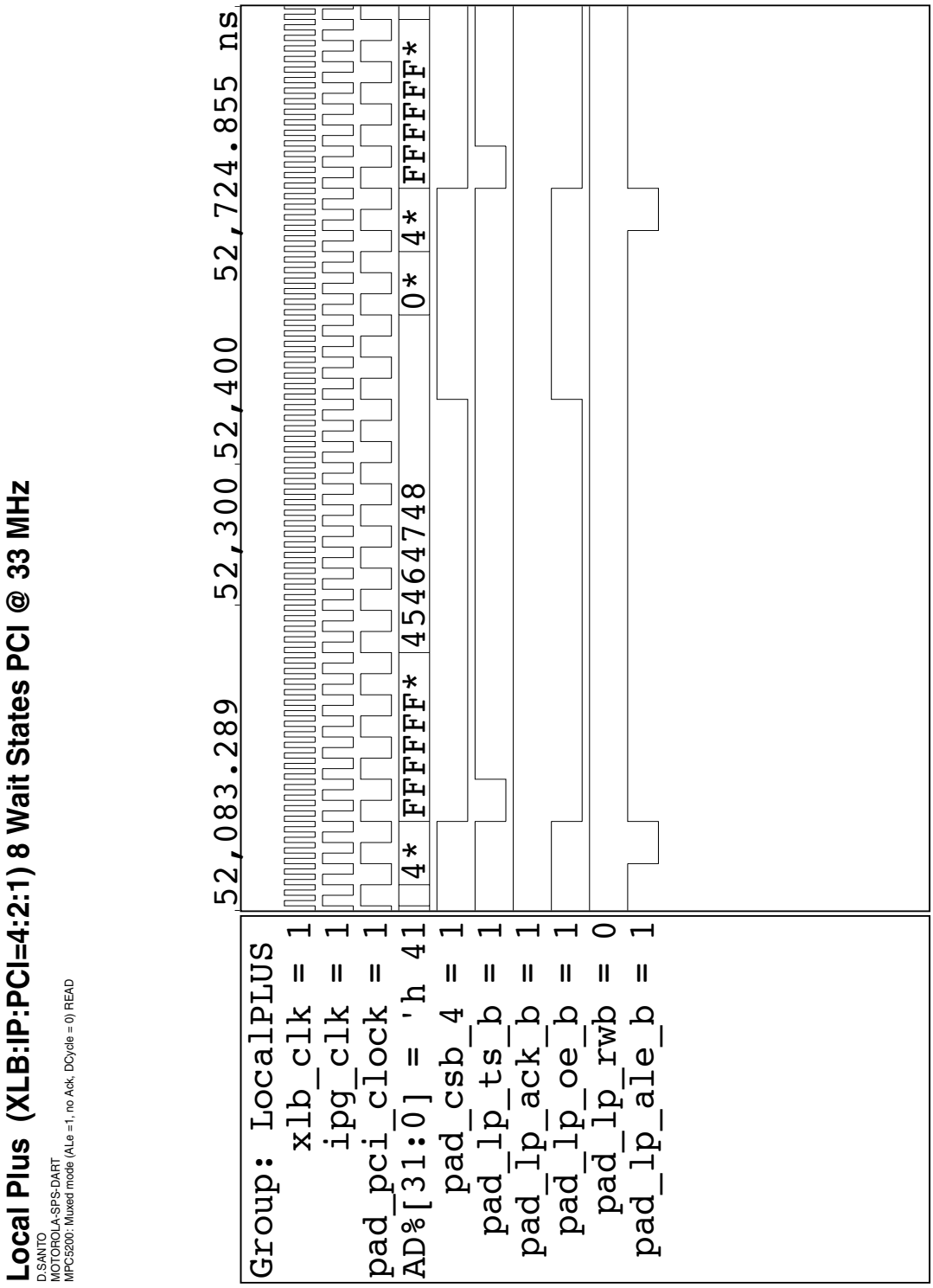


Figure 12. Read Access, 8 Wait States, 0 Dead Cycles

4.6.2 Write Access, 8 Wait States, 0 Dead Cycles

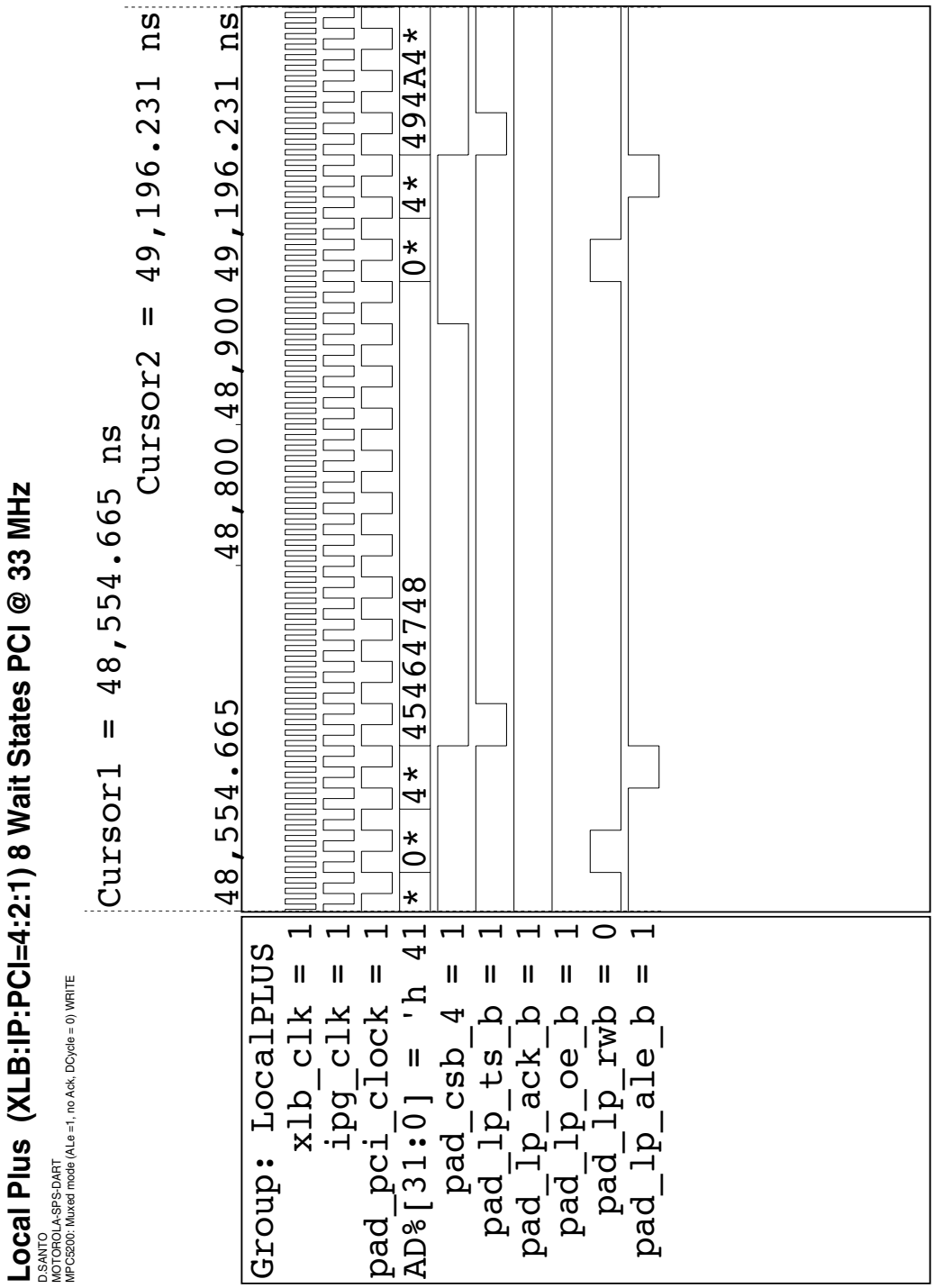


Figure 13. Write Access, 8 Wait States, 0 Dead Cycles

4.6.3 Read Access, 8 Wait States, 2 Dead Cycles

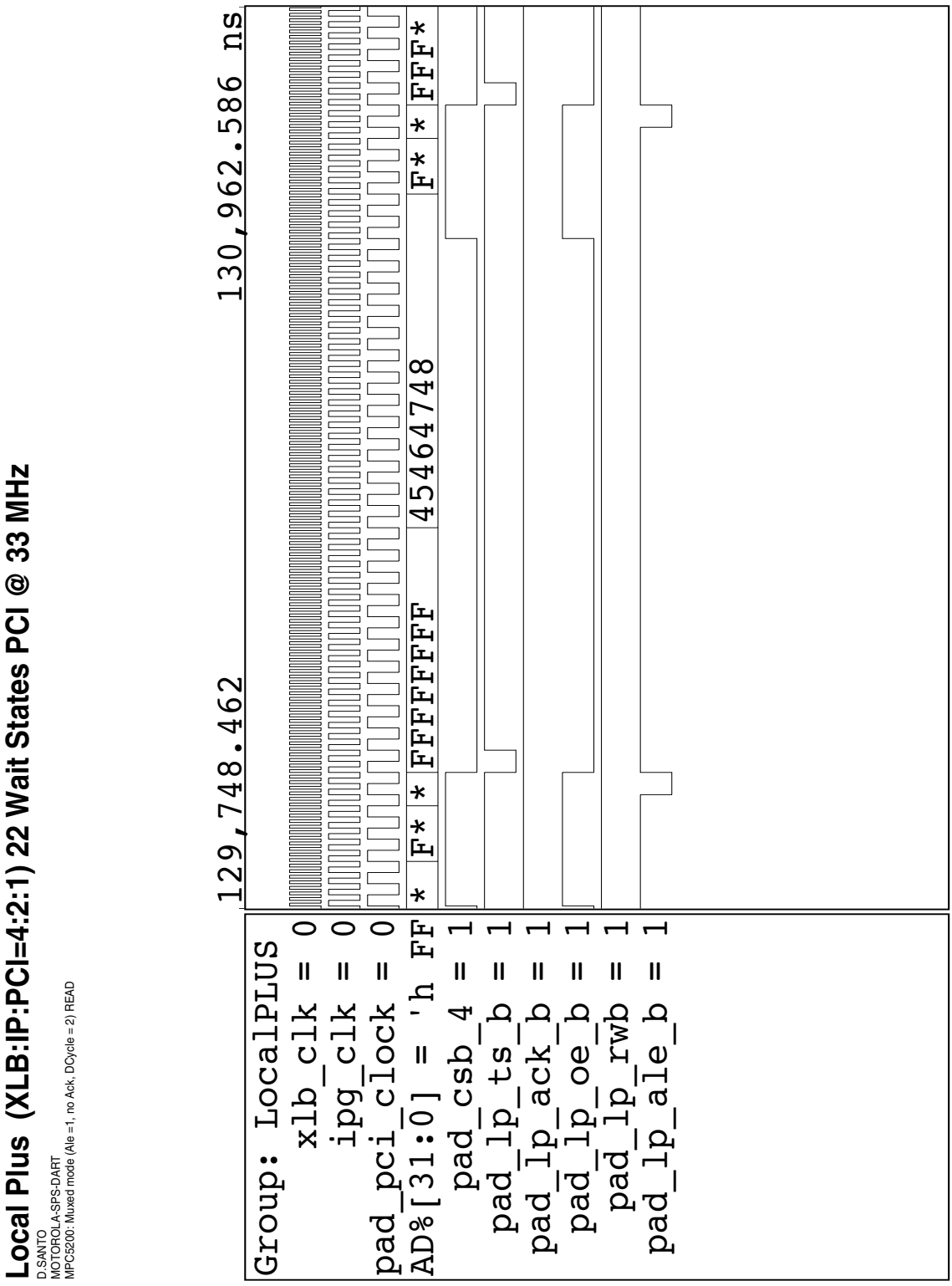


Figure 14. Read Access, 8 Wait States, 2 Dead Cycles

5 BestComm Interface to the Local Plus Bus

The MPC5200 processor has integrated a DMA engine called ***BestComm***. It allows to move data from different peripherals (PSC, PCI, ATA, LP, I2C, ETHERNET) to and from the main memory (SDRAM memory) via FIFOs used as buffers.

Each internally supported peripheral has a specific set of registers to support the DMA process in addition to those needed by the DMA engine to operate (to read about the BestComm Programmer Model please refer to the User Manual section 11).

The Local Plus interface is supported by the DMA engine via a specific interface (called in the User Manual “SCLPC”) using a single 512 byte deep FIFO. This allows the user to implement data transfer either from memory to the Local Plus Bus or from the Local Plus Bus in an ‘half-duplex’ fashion. For sake of completeness, the PCI interface is also supported by BestComm but with 2 FIFOs (one for RX and one for TX) thus enabling a ‘full-duplex’ transfer, while the ATA interface has an ‘half-duplex’ interface to BestComm. Each peripheral has his own FIFO with associated FIFO controller, allowing all three modules (PCI, ATA and LP) to be used at the same time.

The same internal arbiter (called PCI arbiter), which arbitrates the PCI, Local Plus and ATA core originated access, manages the BestComm initiated request (for either PCI, Local plus and ATA).

How to write a BestComm’s task supporting the Local Plus is not treated in this application note. The focus will be more on the steps needed to prepare the Local Plus controller to use the BestComm assuming a proper task capable of moving data to/from the LP bus is available and that the user knows how to enable a task (see User Manual Section 11). The complete description for the SCLPC register can be found in the User Manual at section 9.7.3.

The following is the description of a typical sequence used to program the Local Plus Bus BestComm interface.

5.1 Reset of the FIFO

Before starting any BestComm access, a reset of the FIFO and of the FIFO controller should be performed. This is achieved by setting to 1 the RF and RC bits of the SCLPC Enable register (MBAR+0x3C0C). As long as any of these bits are set hogh no external BEstComm driven operation on the bus is possible. Software must release these bits to allow start of operation (see Clear of reset state, later on).

5.2 Write Start Address Register

The starting address for the transaction must then be set in the SCLPC Start Address Register (MBAR+0x3C04). This can be written even before the reset of the FIFO or FIFO controller without being affected by the latter.

The written address is the same one, which will appear on the Local Plus Bus on the very first BestComm initiated transfer. There are two possibilities then: either the address will be automatically incremented (in case a ‘memory’-like device is accessed) or can be fixed to the initial value (in case a ‘FIFO’-like device is accessed). This can be set by the user by setting the DAI (Disable Auto Increment) bit of the SCLPC Control register.

5.3 Write Control Register

The Control register (MBAR=0x3C08) is then programmed to indicate which chip select (0 through 7) will be used by the BestComm and how many bytes will be moved per transaction (PBT field). The bus port mode (Muxed Mode versus not muxed, data/ address width, ale, ack, number of wait states, etc.) of the bus are still set using the usual control and configuration registers as for the not BestComm initiated accesses. The BPT field can be set from 0 to 7, where ‘0’ means 8 bytes per transaction (1 then meaning 1 byte, and so on). In general the BPT field can be larger than the bus port’s size (example 8 bytes on a 16 bit data port), except in the case where the DAI bit is set (i.e. when writing to an external FIFO, the BPT **must** be of the same size as the FIFO’s port). The BPT can **never** be less than the bus port’s size. The control register RWb field indicates if a WRITE (to the Local Plus) or a READ (from the Local Plus) will be executed. Also the Control register is unaffected by a RESET of the FIFO.

5.4 Clear the FIFO Reset State

It is possible (if not done before) to clear (write to 0) the RF and CF bits in the SCLPC enable register, allowing the FIFO to exit the reset state.

5.5 Write the Interrupt and Master Enable bits

There is a Master enable bit and two interrupt enabling bits in the SCLPC Enable register (MBAR+0x3C0C). The Master bit **must** be set to '1' to permit any operation (and before the first packet is kicked off). The two interrupts bits enable the SCLPC interface to send an interrupt to the core in case of a Normal Termination of a packet (NIE) or an abnormal (i.e., erred) termination (AIE). If an interrupt will be received, the core can read the SCLPC Bytes Status Register (MBAR+0x3C14) to determine whether it was a normal or abnormal termination interrupt signalled.

5.6 Writing the FIFO Watermarks

The Fifo has two 'watermarks' called ***Granularity*** and ***Alarm***. They can be set now to their desired value. ***Granularity*** indicates the 'level' of data in the FIFO at which BestComm will stop operating on it (either filling or emptying it). The ***Alarm*** is the level of data in the FIFO at which BestComm must resume operation and start moving data. The Alarm level can be set to any value by writing the SCLPC "LPC RX/TX FIFO Alarm register" (MBAR+0x3C4C). The Granularity can be set by writing to the SCPLC "LPC RX/TX FIFO Control register" (MBAR+ 0x3C48). Granularity is measured in 32-bit words and can be set from 0 to 7. It is important to remember that BestComm will wait before transmitting to have enough data in the FIFO to cover an entire transaction (whose number of bytes is fixed by the PBT field in the control register). Therefore the Alarm level shall be always be set greater than the PBT size to avoid stalling the DMA engine.

NOTE: DMA request Line

There is no DMA request line available; it is thus NOT possible for an external device to directly initiate a BestComm transfer. It is always the internal core which starts the task.

5.7 Writing the Packet Size Register

Now it is the proper time to write the SCLPC Packet size register (MBAR+0x3C00). This register has 24 bits allocated to indicate the size in bytes of the complete transfer. A single "packet" can then be as large as 65536 bytes. It is important than the Packet Size and the PBT divide evenly (if the PBT is set to 8 then only multiple of 8 bytes are allowed as Packet Size). As long as the transaction is ongoing the next address can be read on the SCLPC NextAddress Status register (MBAR+0x3C10), while the number of bytes transmitted are read from the SCLPC Bytes Done Status Register (lower 24 bits). Resetting the FIFO does NOT alter the Packet Size register.

5.8 Writing the Restart Bit: Kick Off!

Bit number 7 (RESTART bit) of the Packet Size register kicks off the transaction. It is part of the Packet Size register to allow a new packet to be written and started at the same time but the two operations can be performed independently. The RESTART bit always auto clear itself and reads back as zero.

As a complementary information, to avoid stale data during READS (due to the fact that a non-zero Granularity would stop the BsstComm before emptying totally the FIFO) a bit is provided (FLUSH bit) in the SCLPC Control register to ignore the granularity level when the last transaction of a packet is performed.

FIFO status registers are available to gain information about the FIFO state (Overflow, underrun, etc.) in case of errors and during debug phase.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-0047, Japan
0120 191014 or +81 3 3440 3569
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2004. All rights reserved.

AN2458
Rev. 2
08/2004