Integrating an External Bus Interface (PL220) with PL3xx Memory Controllers

Application Note

ARM®

ARM DAI 0184B

Integrating an External Bus Interface (PL220) with PL3xx Memory Controllers
Application Note

Release Information

Table 1 Change history

Date	Issue	Confidentiality	Change
11 May 2007	A	Non-Confidential	First release
15 October 2007	B	Non-Confidential	Addition of *Chip-select timing* on page 8

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

1 About this document

This application note describes how to connect an *External Bus Interface* (EBI) with:

- a PL34x series *Dynamic Memory Controller* (DMC)
- a PL35x series *Static Memory Controller* (SMC).

The application note also contains the following sections:

1.1 Reference

This application note refers to the ARM PrimeCell External Bus Interface (PL220) Technical Reference Manual (ARM DDI 0249).

1.2 Terms and abbreviations

Table 2 lists the terms and abbreviations that this application note uses.

Table 2 Terms and abbreviations

Term	Meaning
DMC	Dynamic Memory Controller
EBI	External Bus Interface
SMC	Static Memory Controller

1.3 Feedback

If you have any comments on this application note, please send e-mail to errata@arm.com giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- an explanation of your comments.

General suggestions for additions and improvements are also welcome.

2 Using EBIGNT

Unlike older PrimeCells, the PL3xx memory controllers do not register **EBIGNT**. This can help to reduce the latency every time the memory bus is granted to a different memory interface.

However, if the EBI is being run at a higher frequency than the memory interface then you must ensure that the rising edge of **EBIGNT** is synchronized to the memory clock.

The handshaking between the EBI and the memory controller consists of a three-wire interface:

- **EBIREQ**
- **EBIGNT**
- **EBIBACKOFF**.

The EBI output, **EBIGNT**, is referred as **EBIGRANT** in PL3xx memory controllers.

3 Synchronization logic

The logic shown in Figure 1 can be used to synchronize **EBIGNT** to the memory clock.

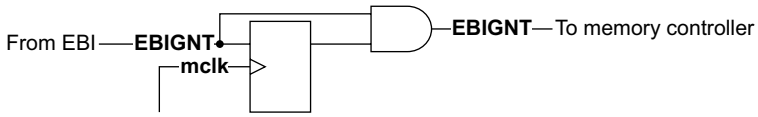

Figure 1 Simple EBIGNT synchronization logic

This ensures that the rising edge of **EBIGNT** is correctly synchronized to **mclk** but leaves the falling edge unchanged and therefore prevents two grants being asserted at the same time.

The logic in Figure 1 will support dynamic changing of **mclk** but will introduce an extra cycle of latency when the EBI is running on the same clock as the memory interface.

Similar logic, but excluding the AND gate, can be used to synchronize the **EBIBACKOFF** signal from the PL3xx series to the EBI. The AND gate is not needed as it is the falling edge on **EBIBACKOFF** that must be synchronized.

Figure 2 shows how the extra cycle of latency can be removed using a general purpose I/O signal, **bypass**, such as one of the user_config signals of PL3xx memory controller.

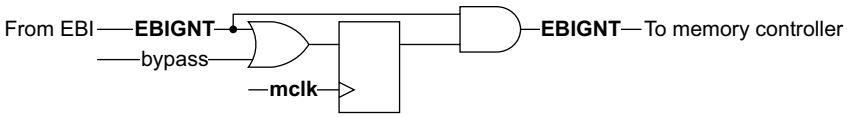

Figure 2 EBIGNT synchronization logic with user signal

The **bypass** control signal must meet the following conditions:

1. The **bypass** signal must only be asserted when the EBI and the memory interface are running at the same frequency.
2. When the **mclk** frequency is being slowed down from the same frequency as **EBICLK** then **bypass** must be LOW before the last rising edge when **EBICLK** matches **mclk**. See Figure 3 and Figure 4.
3. When the **mclk** frequency is being increased to **EBICLK** then **bypass** can be asserted at any point after the last rising edge of **mclk** before **EBICLK** matches **mclk**. See Figure 5 on page 6 and Figure 6 on page 6.

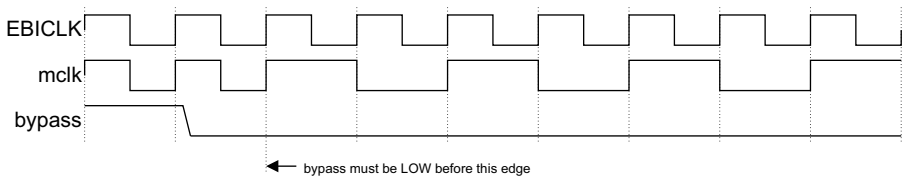

Figure 3 bypass signal timing, reducing frequency 1

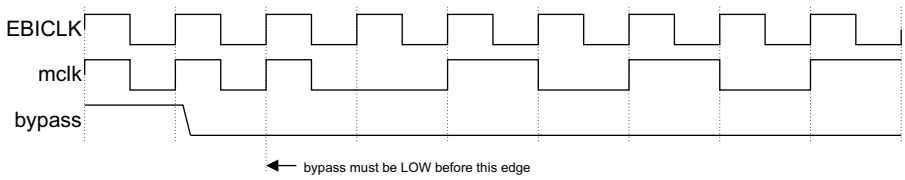

Figure 4 bypass signal timing, reducing frequency 2

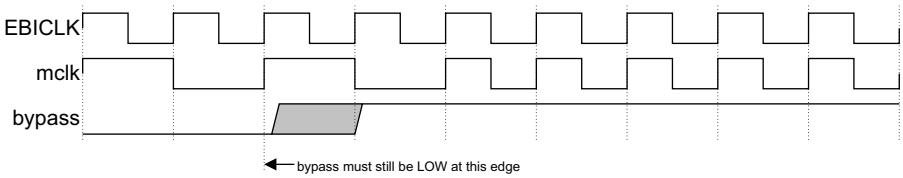

Figure 5 bypass signal timing, increasing frequency 1

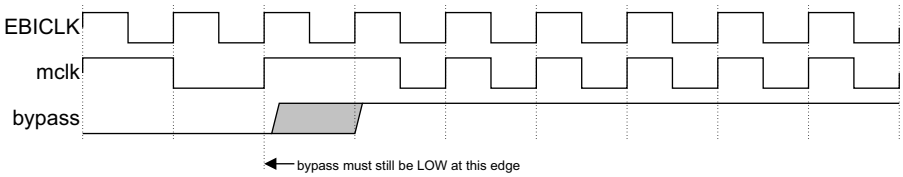

Figure 6 bypass signal timing, increasing frequency 2

4 Clock switching

For the SMC (PL350 series), if the **bypass** signal is kept LOW all the time then it is possible to change clocks as desired if the following conditions are met:

- the clocks remain synchronous.
- **EBICLK** always remains as the fastest of the memory clocks
- the memory timing parameters are never violated.

The same applies when using bypass logic to remove the extra cycle of latency provided that it obeys the rules stated in this application note.

For the DMC (PL340 series), the **EBIBACKOFF** and **EBIGNT** synchronization clocks can also be randomly changed with the same conditions provided that there is no DLL in the system to lose lock.

5 Chip-select timing

If two memory interfaces belonging to the same memory controller or to different ones operate at different frequencies, **mclk1** and **mclk2**, then **EBICLK** runs at the fastest frequency. Figure 7 shows how the chip-selects of the memory interfaces are asserted and deasserted with respect to the EBI protocol timing.

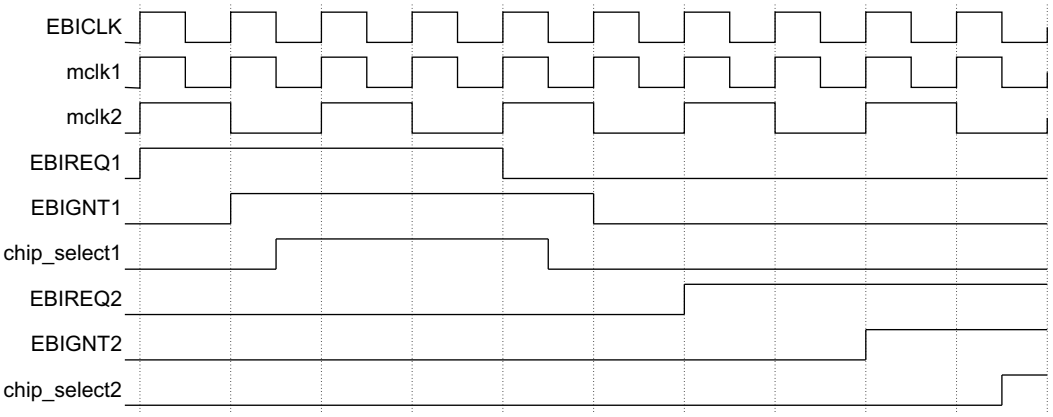

Figure 7 Chip select timing for different memory interface clock speeds

Note

Memory interface 2 has the bypass logic included. Its timing reflects this.