PowerPC™

# Updating PowerPC System Designs From IBM27-82650 Revision 2.0 to Revision 2.1

## Application Note AN-PPC-001

*PowerPC 60X system designs based on revision 2.0 of the IBM27–82650 Bridge can be upgraded to use revision 2.1 of the bridge by removing external logic.*

Dale Elson
February 1995

IBM®

# Updating PowerPC System Designs From the IBM27–82650 Revision 2.0 to Revision 2.1

Dale Elson

*PowerPC 60X system designs based on revision 2.0 of the IBM27-82650 Bridge can be upgraded to use revision 2.1 of the bridge by removing external logic.*

## Table of Contents



## 1 Overview

Various PowerPC 60X system designs have been developed that are based on revision 2.0 of the IBM27-82650 PowerPC to PCI Bridge. This document describes the changes that are necessary to upgrade these system designs for compatibility with revision 2.1 of the bridge. For each topic in the following discussion, the original errata, the level 2.0 fix, and the level 2.1 fix are described.

Bridge signal names referenced in the equations include the chip and pin number in brackets following the signal name. For example, the signal named NO_TRANS[654.13] connects to the 654 Controller at pin 13.

Some systems may have been designed using one or more PLD devices to implement the required external logic. All such devices are collectively referred to below as the PLD. The equations shown in the following errata are presented in the form of PLD compiler input.

Note that all errata have been corrected in the 2.1 revision of the bridge. Therefore system designs based on the revision 2.1 bridge do not require the external logic that was required to correct the errata associated with revision 2.0 designs. When removing the external logic from the design, be sure to observe correct physical design rules for high speed logic by removing any associated trace stubs.

## 2 ISA Master Support for SIO Posted Writes (Level 2.0 Errata 1)

ISA master detection and address translation support for Intel® 82378 PCI to ISA bridge (SIO) does not correctly detect an ISA write to system memory performed from the posted write buffer within the SIO. Also, an arbitration cycle that results in a PCI bus grant to the SIO device to service an ISA master cycle before completion of a previous PCI burst cycle can result in a system error.

### 2.1 Level 2.0 Solution

An external logic workaround has been designed and tested. However, the SIO is only allowed to master transactions to system memory (SIO response as a PCI target is unaffected). The SIO may not master transactions to other PCI agents.

### 2.2 Level 2.0 Implementation

Cut the connection between NO_TRANS[653.43] and NO_TRANS[654.13]. Cut the connection between PCI_FRAME#[654.29] and the PCI bus signal FRAME#. Connect BUS_FRAME# (below) to the PCI bus signal FRAME# (or internally to the PLD term in errata 4).

PLD Logic Equations:

FRAMEL#.D = BUS_FRAME#
FRAMEL_.CLK = PCI_CLK

Q_SIO_GNT.D = $\overline{\text{IO\_BRDG\_GNT\#[654.7]}}$ * BUS_FRAME#
* PCI_IRDY#[654.52]
+ Q_SIO_GNT.Q * $\overline{\text{BUS\_FRAME\#}}$
+ Q_SIO_GNT.Q * $\overline{\text{PCI\_IRDY\#[654.52]}}$
Q_SIO_GNT.CLK = PCI_CLK

NO_TRANS[653.43].D = $\overline{\text{BUS\_FRAME\#}}$ * PCI_IRDY#[654.52]
* Q_SIO_GNT.Q * $\overline{\text{AD31[653.61]}}$
* FRAMEL#.Q
+ NO_TRANS[653.43].Q * $\overline{\text{PCI\_IRDY\#[654.52]}}$
+ NO_TRANS[653.43].Q * $\overline{\text{BUS\_FRAME\#}}$
NO_TRANS[653.43].CLK = PCI_CLK

### 2.3 Level 2.1 Solution

The function of the ISA_MASTER# pin has been changed for revision 2.1 of the chipset. Connect the 654 Controller ISA_MASTER# pin to a copy of PCI_AD[31] (see below). During PCI bus memory transactions mastered by the ISA bridge, ISA_MASTER# is sampled on the rising edge of the PCI_CLK on which PCI_FRAME# is first sampled active and is required to be valid at that time.

Table 1 shows the action taken by the 650 Bridge during PCI bus memory transactions mastered by the ISA bridge. If ISA_MASTER# is low (the PCI bus address is from 0 to 2G), the 654 Controller asserts NO_TRANS to the 653 Buffer. The PCI memory address is not translated by the 650 Bridge and is broadcast to the CPU bus for snooping. A system memory access in the 0 to 2G range is generated by the 650 Bridge.

**Table 1. ISA_MASTER# Operation**

| ISA_MASTER# (Address Phase) | PCI Bus Address Range | NO_TRANS | CPU Bus and Memory Address | Transaction Target |
|---|---|---|---|---|
| 0 | 0 to 2G | 1 | 0 to 2G | System Memory |
| 1 | 2G to 4G | 0 | 0 to 2G | System Memory |

If the ISA bridge memory transaction is to the 2G to 4G address range, then ISA_MASTER# is high. The 654 negates NO_TRANS to the 653, and the 653 translates the address to the 0 to 2G range for the memory access and snoop.

This forces all ISA bridge-mastered PCI memory transactions to access system memory. Therefore, the ISA bridge is not permitted to master PCI memory transactions to other PCI agents. ISA_MASTER# only affects PCI memory transactions mastered by the ISA bridge. PCI I/O transactions and transactions by other bus masters are not affected.

This information replaces the information about the ISA_MASTER# signal contained in the 650 Bridge Users Manual on pages 24, 31, 98, 249, and 259.

## 2.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

The MASTER# pin of the ISA bridge should be connected according to the ISA bridge documentation and as shown on page 22 of the reference system schematic. Disconnect this signal from the 650 Bridge. Connect ISA_MASTER#[654.79] to a buffered copy of PCI_AD[31]. ISA_MASTER#[654.79] requires a setup time to the PCI_CLK of 7ns and a 0ns hold time, and the signal is sampled during the address phase of a PCI transaction. Note that ISA_MASTER#[654.79] has an absolute maximum input voltage range of -.5v to +5.5v. The reference design connects ISA_MASTER#[654.79] to PCI_AD[31] through a resistor with a small enough value that ISA_MASTER#[654.79] is forced to meet the setup time to the PCI_CLK and large enough to avoid adding a load to the PCI bus—but buffering is recommended.

Disconnect NO_TRANS[653.43] from the PLD.
Connect NO_TRANS[653.43] to NO_TRANS[654.13].

Disconnect PCI_FRAME#[654.29] from the PLD.
Disconnect the PCI bus FRAME# signal from the PLD.
Connect PCI_FRAME#[654.29] to the PCI bus FRAME# signal.

PCI_IRDY#[654.52] is no longer required by the PLD.
PCI_AD[31] is no longer required by the PLD.
PCI_CLK is no longer required by the PLD.
IO_BRDG_GNT#[654.7] is no longer required by the PLD.

## 3 CPU Bus Arbitration (Level 2.0 Errata 2)

The 654 Controller does not detect the deassertion of the 60X CPU bus request signal (CPU_REQ#) if the deassertion is only one CPU_CLK in duration and occurs while PCI_CLK is high. This errata is only of concern if CPU_CLK is running at twice the frequency of PCI_CLK (for example, 66MHz CPU_CLK and 33MHz PCI_CLK). The net effect is that extended bus grants to the CPU can increase PCI bus latency. In extreme cases, buffer overruns of PCI bus masters can occur due to lack of available PCI bus ownership.

### 3.1 Level 2.0 Solution

External logic has been designed and tested which extends the minimum CPU_REQ# high pulse width (at the 654 Controller input) to two CPU bus clocks. This has minimal impact on performance.

### 3.2 Level 2.0 Implementation

Cut the connection between the 60X CPU bus signal $\overline{\mathrm{BR}}$ (CPU_BR#) and CPU_REQ#[654.126].

PLD Logic Equations:

BR_L.D = CPU_BR#
BR_L.CLK = CPU_CLK[654.142]

BR_L2.D = BR_L.Q
BR_L2.CLK = CPU_CLK[654.142]

CPU_REQ#[654.126] = BR_L.Q + BR_L2.Q

### 3.3 Level 2.1 Solution

The level 2.0 solution has been incorporated into the 654 Controller logic.

### 3.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

Disconnect CPU_REQ#[654.126] from the PLD.
Disconnect the CPU bus BR# signal from the PLD.
Connect the CPU bus BR# signal to CPU_REQ#[654.126].
CPU_CLK is no longer required by the PLD.

# 4 PCI Pacing of System Memory Burst Writes (Level 2.0 Errata 3)

The memory controller in the 654 Controller does not recognize an attempt by a PCI bus master to pace a burst write to system memory. Memory is corrupted if the bus master holds PCI_IRDY# deasserted for more than one PCI bus clock during a data phase of a burst write.

## 4.1 Level 2.0 Solution

External logic has been designed and tested which causes the 654 Controller to retry the PCI bus master if PCI_IRDY# remains deasserted for more than one PCI bus clock. Memory is not corrupted.

## 4.2 Level 2.0 Implementation

Leave ARTRY#[654.112] connected to the CPU bus signal $\overline{\text{ARTRY}}$.

PLD Logic Equations:

ARTRY#[654.112].D = logic zero
ARTRY#[654.112].OE = ARTRY_ENB.Q

ARTRY_ENB.D = (ASET.Q * $\overline{\text{CPU\_ADDR\_OE\#[654.144]}}$ * $\overline{\text{TT1[654.122]}}$
* $\overline{\text{AACK\#[654.113]}}$ * PCI_IRDY#[654.52])
ARTRY_ENB.CLK = CPU_CLK[654.142]

ASET.D = ($\overline{\text{CPU\_ADDR\_OE\#[654.144]}}$ * $\overline{\text{AACK\#[654.113]}}$)
+ (ASET.Q * $\overline{\text{PCI\_FRAME\#[654.29]}}$)
+ (ASET.Q * $\overline{\text{PCI\_IRDY\#[654.52]}}$)
+ (ASET.Q * PCI_CLK[654.129])
ASET.CLK = CPU_CLK[654.142]

## 4.3 Level 2.1 Solution

The level 2.0 solution has been incorporated into the 654 Controller logic.

## 4.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

Leave ARTRY#[654.112] connected to the CPU bus signal $\overline{\text{ARTRY}}$.
Disconnect the PLD from ARTRY#[654.112] and the CPU bus signal $\overline{\text{ARTRY}}$.

CPU_ADDR_OE#[654.144] is no longer required by the PLD.
TT1[654.122] is no longer required by the PLD.
AACK#[654.113] is no longer required by the PLD.

# 5 Final Data Phase of a PCI to Memory Read (Level 2.0 Errata 4)

The snoop cycle generator in the 654 Controller does not generate a snoop cycle to the CPU for the final data phase of a PCI burst read from system memory. If this last data beat is the first word in a new cache line and the cache line is dirty, the 654 Controller returns stale data to the PCI master.

## 5.1 Level 2.0 Solution

External logic has been designed and tested that lengthens the PCI_FRAME# signal into the 654 Controller, causing the snoop cycle to be generated. Lengthening PCI_FRAME# does not otherwise affect the cycle in progress.

### 5.2 Level 2.0 Implementation

PLD Logic Equations:

INT_FRAME#.D = PCI_TRDY#[654.27] + BUS_FRAME#
INT_FRAME#.CLK = PCI_CLK[654.129]

PCI_FRAME#[654.29] = BUS_FRAME# * INT_FRAME#.Q
PCI_FRAME#[654.29].OE = $\overline{\text{CPU\_ADDR\_OE\#[654.144]}}$

BUS_FRAME# = PCI_FRAME#[654.29]
BUS_FRAME#.OE = CPU_ADDR_OE#[654.144]

### 5.3 Level 2.1 Solution

The level 2.0 solution has been incorporated into the 654 Controller logic.

### 5.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

Disconnect PCI_TRDY#[654.27] from the PLD.

## 6 MC_SETUP# Race Condition (Level 2.0 Errata 5)

The current 650 Bridge User’s Manual lists no setup or hold (to CPU_CLK) specification for MC_SETUP#. However, at the end of a 654 Controller control register programming cycle, the rising edge of MC_SETUP# cannot coincide with the rising edge of CPU_CLK, or the CAS# generation logic may not properly reset, thereby causing memory problems.

### 6.1 Level 2.0 Solution

The system designer must ensure that MC_SETUP# does not transition coincident with the CPU bus clock rising edge. Simulation has shown that a minimum setup time of 2ns and a minimum hold time of 1ns from MC_SETUP# transitions to the rising edge of CPU_CLK are sufficient to prevent memory failures.

### 6.2 Level 2.0 Implementation

The system designer guarantees the required minimum setup and hold times.

### 6.3 Level 2.1 Solution

This input has been made completely asynchronous. There is no setup or hold time required from this signal to any clock.

### 6.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

Circuitry used to synchronize MC_SETUP# to the CPU_CLK is no longer required.

# 7 CPU Bus Parity Transceiver Enables (Level 2.0 Errata 6)
On CPU read cycles that hit in the L2 cache, the L2 cache drives 72 bits (64 data bits plus 8 parity bits). The 654 Controller also asserts the CPU bus parity transceiver enables for one CPU bus clock, causing a bus contention.

## 7.1 Level 2.0 Solution
External logic has been designed and tested which prevents the bus contention and allows the use of an L2 cache.

## 7.2 Level 2.0 Implementation
Cut the connection between LE_PAR_EN#[654.85] and the little-endian parity buffer. Cut the connection between BE_PAR_EN#[654.85] and the big-endian parity buffer. Connect LE_PAR_EN# to the output enable pin of the little-endian parity buffer. Connect BE_PAR_EN# to the output enable pin of the big-endian parity buffer.

PLD Logic Equations:

LE_PAR_EN# = LE_PAR_EN#[654.85] + CAS0#[654.22]

BE_PAR_EN# = BE_PAR_EN#[654.84] + CAS0#[654.22]

## 7.3 Level 2.1 Solution
The level 2.0 solution has been incorporated into the 654 Controller logic.

## 7.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset
Disconnect the PLD from the little-endian parity buffer.
Disconnect the PLD from the big-endian parity buffer.
Connect LE_PAR_EN#[654.85] to the little-endian parity buffer.
Connect BE_PAR_EN#[654.85] to the big-endian parity buffer.
CAS0#[654.22] is no longer required by the PLD.

# 8 RAS Minimum Pulse Width (Level 2.0 Errata 7)
Under worst case conditions, the minimum RAS# pulse width can be as short as 69ns for memory refresh cycles.

## 8.1 Level 2.0 Solution
None.

## 8.2 Level 2.0 Implementation
None.

## 8.3 Level 2.1 Solution
The RAS timing has been improved to at least 70ns.

## 8.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset
No action required.

# 9 Considerations for use with 32M SIMMs (Level 2.0 Errata 8)

Under certain conditions, the 653 Buffer produces a multiple state change pattern on the memory address lines (while RAS# is low and stable) that causes certain 32-Mbyte SIMMs to malfunction. The following solution prevents the SIMM malfunction by forcing the RAS# lines inactive over the interval during which the memory address line slope reversals may be present.

## 9.1 Level 2.0 Solution

If SIMMs with this previously undocumented sensitivity are present in the system, force single-beat (or data phase) memory accesses and the first beat (or data phase) of burst memory accesses to be a page miss. This forces correct information into the DRAM row address latches. As implemented below, subsequent beats (or data phases) of burst memory transactions are page hits as long as A[19] (big-endian) does not change state (which would indicate the crossing of a 4K page boundary).

## 9.2 Level 2.0 Implementation

If required, disconnect MEM_PAGE_HIT#[654.149] from MEM_PAGE_HIT#[653.64],
Tie MEM_PAGE_HIT#[654.149] high.

## 9.3 Level 2.1 Solution

Same as the level 2.0 solution.

## 9.4 Updating Level 2.0 Planars to use Revision 2.1 of the Chipset

No action required.

PowerPC



**USA and Canada:**
IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6531
Tel: (800) PowerPC
Fax: (800) PowerFax

**Europe:**
IBM
La Pompignane BP 1021
34006 Montpellier, France
Tel: (33) 6713–5757 (Français)
(33) 6713–5756 (Italiano)
Fax: (33) 6713–5750
( from Paris add 16)

**Europe:**
IBM
Informations Systeme GmbH
Laatzener Str. 1
30539 Hannover, Germany
Tel: (49) 511–516–3444 (English)
(49) 511–516–3555 (Deutsche)
Fax: (49) 511–516–3888

**Japan:**
IBM
800, Ichimiyake
Yasu–cho, Yasu–gun
Shiga–ken, Japan 520–23
Tel: (81) 775–87–4745
Fax: (81) 775–87–4735

Document Number: MPRA01APU–01