PCI 9054/MCF5307 AN

July 7, 2000

Version 2.0

MCF5307 to PCI 9054 Application Note

Features

- Application Note for designing a PCI adapter or embedded system based on the Motorola MCF5307 ColdFire microprocessor and the PLX PCI 9054
- Superior PCI performance based on the PCI 9054 bus master interface chip which supports:
  - PCI burst master, DMA and slave cycles
  - PCI configuration cycles
  - I_2O^{TM} Messaging Unit
  - CompactPCI Hot Swap

General Description

This application note describes how to interface the Motorola MCF5307 microprocessor to the PCI bus using the PLX PCI 9054 32-bit I/O Accelerator device. The information can be used to build either a PCI adapter or a host.

The PCI 9054 has Direct Master, DMA and Direct Slave data transfer capabilities. The Direct Master mode allows a device (MCF5307) on the Local bus to perform memory, I/O, and configuration cycles to the PCI bus. The Direct Slave gives a master device on the PCI bus the ability to access memory on the Local bus. The PCI 9054 allows the Local bus to run asynchronously to the PCI bus through the use of bi-directional FIFOs. In this design example, the PCI bus runs at 33 MHz, while the Local bus can be clocked at up to 45 MHz (an MCF5307 limitation).

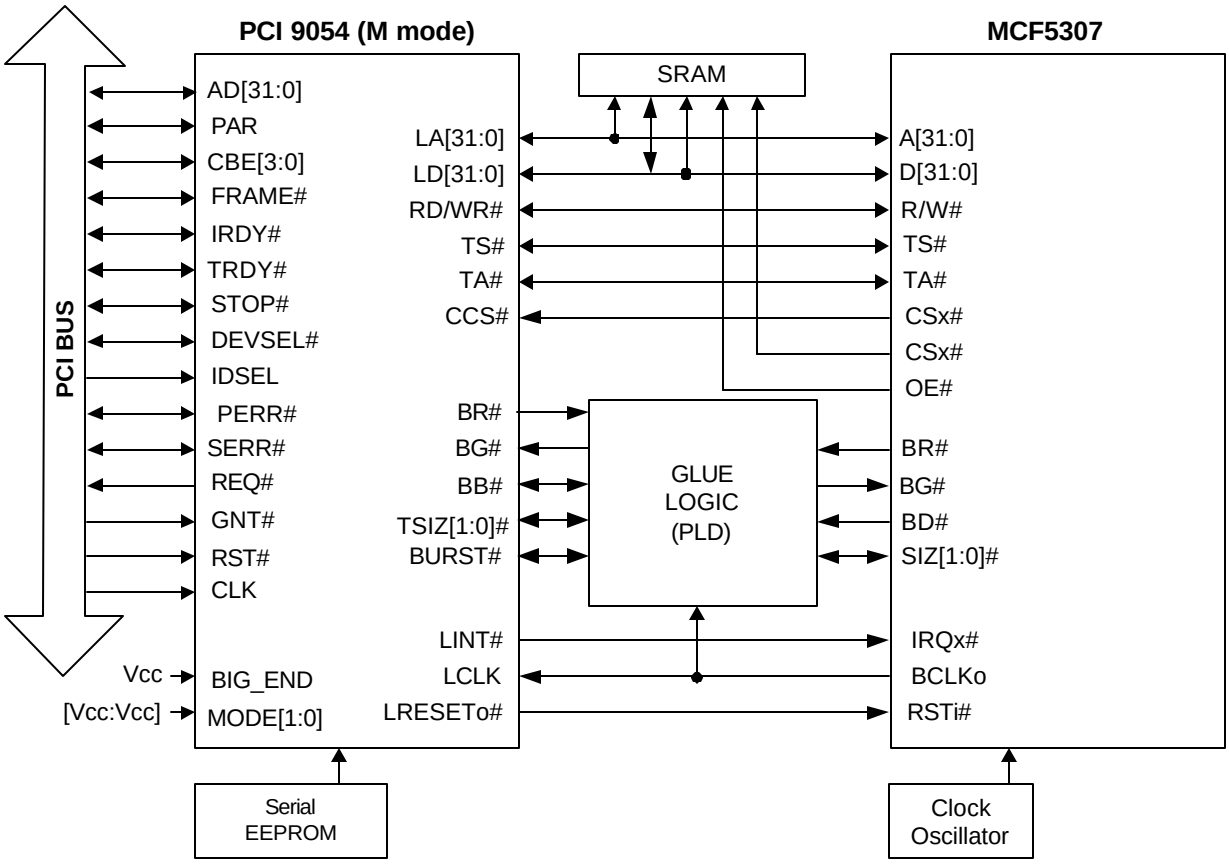

Figure 1. MCF5307 to PCI Subsystem

PLX Technology, Inc, 390 Potrero Avenue, Sunnyvale, CA 94085, Phone 408-774-9060, Fax 408-774-2169

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

This application note describes the interconnect between the Motorola MCF5307 ColdFire and the PCI bus using the PLX PCI 9054 "PCI to Local bus bridge" IC. This application note can be used as a basis to build either a PCI adapter or host to plug into a PCI bus backplane, or for building an embedded system. The interconnect between the PCI 9054 and the MCF5307 also requires the use of extra logic, in the form of a PLD or other programmable logic. Figure 1 shows the connections between the PCI 9054, the MCF5307 and the PLD. Figure 2 shows the detail in the PLD.

The PCI 9054 has Direct Master (DM), DMA and Direct Slave (DS) transfer capabilities. The Direct Master mode allows the MCF5307 to perform memory, I/O and configuration cycles to the PCI bus.

The PCI 9054 also contains a powerful chaining DMA controller. Although it is not discussed in this design example, it can be used with no changes to the hardware. For long and efficient burst transfers between a PCI host and an adapter's memory, the DMA mode should be considered.

The Direct Slave mode gives a master device on the PCI bus the ability to access the MCF5307 configuration registers or memory on the Local bus. This allows for burst or single cycle direct slave transfers.

The PCI 9054 allows the Local bus to run asynchronously to the PCI bus through the use of bi-directional FIFOs. In this application, the PCI bus runs at 33 MHz while the Local bus is clocked at 45 MHz.

2. ARCHITECTURE

The Block Diagram of the PCI 9054/MCF5307 application (Figure 1) will be discussed in the following subsections. The Local bus consists of all the signals in the application block diagram that are connected to the MCF5307. Only the PCI 9054 and the MCF5307 can be the Local bus masters. The MCF5307 can access the PCI 9054 registers as well. The PCI 9054 and MCF5307 are both configured for Big Endian operation. The PCI 9054 has the capability of byte swapping Big Endian data into Little Endian data, which can then be sent over to the PCI bus.

The PCI 9054 to MCF5307 interconnect is straightforward, with a single PLD used to translate a few signals between the two devices. One function that is performed in the PLD is to provide Local bus arbitration. Another function that must be performed is adapting transfer size signals.

2.1 PCI 9054 Bus Mode Used

To connect the PCI 9054 to the MCF5307, the M-Bus mode was chosen. This is due to the fact that the MCF5307 32-bit bus interface is essentially an M-mode interface, similar to the MPC860.

2.2 Control Signal Connections

Below is a list of the PCI 9054 signals, their corresponding MCF5307 signals [], and a brief description of what they do:

LA[31:0] - (A[31:0]) - Carry the 32-bits of the physical Address bus. LA0 is the most significant bit of the bus address. Connected directly to MCF5307 A[31:0] lines.

LD[31:0] - (D[31:0]) - Carries 8-, 16-, or 32-bit data quantities, depending upon the bus-width configuration. All master accesses to the PCI 9054 are 32-bits only. LD0 is the most significant bit of the bus address. Connected directly to MCF5307 D[31:0] lines.

RD/WR# - (R/W#) - Asserted high for reads and low for writes. Connected directly to the MCF5307 R/W# line.

TS# - (TS#) - Indicates the valid address and start of a new bus access. Asserted for the first clock of a bus access. Connected directly to the MCF5307 TS# line.

TA# - (TA#) - As an input, when a Local bus access is made to the PCI 9054, indicates a write data transfer can complete or that read data on the bus is valid. As an output, when the PCI 9054 is a bus master, indicates a write data transfer is complete or that read data on the bus is valid. Connected directly to MCF5307 TA# line.

CCS# - (CSx#) - Internal PCI 9054 registers are selected when CCS# is asserted low. Connected directly to one of the MCF5307 CS outputs.

BR# - Asserted by the master to request use of the Local bus. The Local bus arbiter asserts BG# when the master is next in line for bus ownership. Connected to MCF5307 and PCI 9054 BG# inputs through glue logic.

BG# - Asserted by the Local bus arbiter in response to BR#. Indicates the requesting master is next. Controlled by glue logic.

BB# - As an input, monitors this signal to determine whether the external master has ended a bus cycle. As an output, the PCI 9054 asserts this signal after an external arbiter has granted ownership of the Local bus and BB# is inactive from another master. Signal requires an external pull-up resistor value of 510OHms to guarantee a fast transition to the inactive state when the PCI 9054 relinquishes ownership of the Local bus. Used in glue logic to control BG# inputs of MCF5307 and PCI 9054.

TSIZ[1:0] - Driven by the current master along with the address, indicating the data-transfer size. TSIZ0 is most significant bit of the bus address. Connected to SIZ[1:0] lines of MCF5307 through glue logic.

BURST# - As an input, driven by the master along with address and data is indicating that a burst transfer is in progress. As an output, driven by the PCI 9054 along with address and data is indicating that a burst transfer is in progress. Used in transfer size subsection of glue logic.

LINT# - (IRQx#) - As an input, when asserted low, causes PCI interrupt. As an output, a synchronous level output that remains asserted as long as an interrupt condition exists. If edge level interrupt is required, disabling and then enabling Local interrupts through INTCSR creates an edge if an interrupt condition still exists or a new interrupt condition occurs. Connected directly to one of MCF5307 IRQ inputs.

LCLK – (BCLKo) - Local clock input. Connected directly to MCF5307 BCLKo clock output.

LRESETo# - (RSTi#) - Asserted when the PCI 9054 chip is reset from the PLX side. Connected directly to RSTi# input of MCF5307.

2.3 Programmable Logic

Although both the PCI 9054 and the MCF5307 have M-mode interfaces, there are enough small differences between the two devices to warrant a PLD to translate some of the signals. The PLD accomplishes two tasks:

1. Arbitrates Local bus to PCI 9054 to enable it to access local memory, and returns bus control to MCF5307 after that.
2. Converts PCI 9054 transfer size signals to MCF5307 transfer size signals.

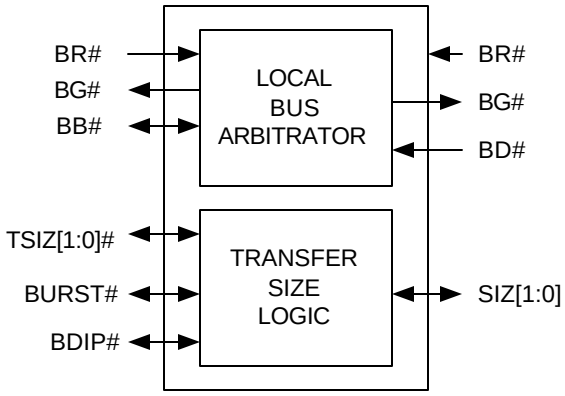

Figure 2. PCI 9054 to MCF5307 PLD

For this a small PLD is needed, such as Xilinx CPLD XC9536XL-5VQ64C or a Lattice 22V10 PAL. The VHDL code is given in the Appendix.

3. ASSUMPTIONS

This application note is based on the following assumption:

- Some typically necessary design components (i.e. boot and code memory, EEPROM, and pull-up/down resistors) are not included in this application note.

The designer is expected to add such components, as needed.

4. REFERENCES

- PLX PCI 9054 Data Book v2.1
  PLX Technology, Inc.
  390 Potrero Avenue
  Sunnyvale, CA 94085 USA
  Tel: 408 774-9060, 800 759-3735
  Fax: 408 774-2169
  http://www.plxtech.com

- MCF5307 User's Manual
  Motorola, Inc.
  1150 Kifer Rd.
  Sunnyvale, CA USA
  Tel: 408 749-0510
  http://www.motorola.com
  Motorola, Inc.

A. APPENDIX

VHDL code for glue logic:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY glue IS
  PORT
  (
    LCLK :  IN STD_LOGIC;
    BRM_N  :  IN STD_LOGIC;
    BGM_N  :  OUT  STD_LOGIC;
    BDM_N  :  IN STD_LOGIC;
    SIZ1_N : INOUT STD_LOGIC;
    SIZ0_N : INOUT STD_LOGIC;
    BRP_N  :  IN STD_LOGIC;
    BGP_N  :  OUT  STD_LOGIC;
    BBP_N  : IN STD_LOGIC;
    TSIZ1_N : INOUT STD_LOGIC;
    TSIZ0_N : INOUT STD_LOGIC;
    BURST_N : INOUT STD_LOGIC
  );
END glue;
ARCHITECTURE a OF glue IS
  SIGNAL Q1 : STD_LOGIC;
  SIGNAL Q2 : STD_LOGIC;
  SIGNAL Q3 : STD_LOGIC;
BEGIN
-- Local bus arbitration
bgm3ff:  PROCESS (LCLK, Q3)
  BEGIN
    IF (Q3 = '1') THEN
      Q1 <= '0';
    ELSIF (LCLK'EVENT AND LCLK = '1') THEN
      IF BRP_N = '0' THEN
        Q1 <= '1';
      ELSE
        Q1 <= Q1;
      END IF;
    END IF;
  END PROCESS;
rst1ff:  PROCESS (LCLK, Q3)
  BEGIN
    IF (Q3 = '1') THEN
      Q2 <= '0';
    ELSIF (LCLK'EVENT AND LCLK = '1') THEN
      IF BBP_N = '0' THEN
        Q2 <= '1';
      ELSE
        Q2 <= Q2;
      END IF;
    END IF;
  END PROCESS;
```

```
rst2ff:   PROCESS (LCLK, BRP_N)
  BEGIN
    IF (BRP_N = '0') THEN
      Q3 <= '0';
    ELSIF (LCLK'EVENT AND LCLK = '1') THEN
      IF Q2 = '1' THEN
        Q3 <= BBP_N;
      ELSE
        Q3 <= Q3;
      END IF;
    END IF;
  END PROCESS;
BGM_N <= Q1;
BGP_N <= NOT BDM_N;
-- Transfer size logic
tsl: PROCESS (BURST_N, TSIZ1_N, TSIZ0_N, SIZ1_N, SIZ0_N)
  BEGIN
    IF (BURST_N = '0') THEN
      SIZ1_N <= '1';
      SIZ0_N <= '1';
    ELSE
      SIZ1_N <= TSIZ1_N;
      SIZ0_N <= TSIZ0_N;
    END IF;
    IF (SIZ1_N = '1' AND SIZ0_N = '1') THEN
      BURST_N <= '0';
      TSIZ1_N <= '0';
      TSIZ0_N <= '0';
    ELSE
      BURST_N <= '1';
      TSIZ1_N <= SIZ1_N;
      TSIZ0_N <= SIZ0_N;
    END IF;
  END PROCESS tsl;
END a;
```