Freescale Semiconductor
Application Note

AN3223
Rev. 0, 2/2006

MPC5200 ATA Interface

by: Martin Nykodym
Transportation Standard Products Group

1 Introduction

This application note describes initialization, general rules, programming model and performance analysis for the MPC5200 ATA interface, mainly from a software perspective. The main focus is to help system architects understand which mode under what circumstances is optimal for the final product. Detailed software examples can be found on the publicly available CD with sample code for the MPC5200. The MPC5200B microcontroller is based on an e300[1] core using the PowerPC™ instruction set.

The ATA interface is used to connect:

- Hard disks
- CD-ROMs
- DVDs
- Flash storage devices

The MPC5200 ATA interface is fully compatible with ATA/ATAPI-4 specification (*AT Attachment with Packet Interface Extension,* ANSI NCITS 317-1998) supporting all three groups of modes:

1. e300, 603e, and G2LE are synonymous.

Table of Contents

freescale™ semiconductor

- PIO (from PIO-0 to PIO-4) – up to 16.7 MBytes/sec
- Multiword DMA (from MDMA-0 to MDMA-2) – up to 16.7 MBytes/sec
- Ultra DMA (from UDMA-0 to UDMA-2) – up to 33 MBytes/sec

MPC5200 supports 27-bit and 48-bit LBA addressing on every device.

2 ATA Interface

The MPC5200 ATA interface includes three groups of registers:

- ATA host registers – host configuration, timing values for all modes (from PIO, MDMA, UDMA)
- ATA FIFO registers – control ATA FIFO for DMA modes (MDMA, UDMA)
- ATA drive registers – access to the registers physically located on the ATA drive

Figure 1 illustrates basic components of the MPC5200 ATA interface internal to the MPC5200. Data transfer is driven either by MPC5200 core (PIO) or by the BestComm DMA engine (MDMA, UDMA). BestComm is designed to offload the MPC5200 core and can transfer data to/from different peripherals simultaneously. The MPC5200 ATA interface is clocked by the IPBI clock of the MPC5200 processor (66/132 MHz). MPC5200 acts as an ATA host and can control up to two ATA devices as defined in ATA/ATAPI-4 specification. Pin ATA_ISOLATION of the MPC5200 (not defined as a signal by ATA/ATAPI-4 spec) connects to the transceiver's OE pin to control the direction (high = write to drive, low = read from drive) of the transfer.

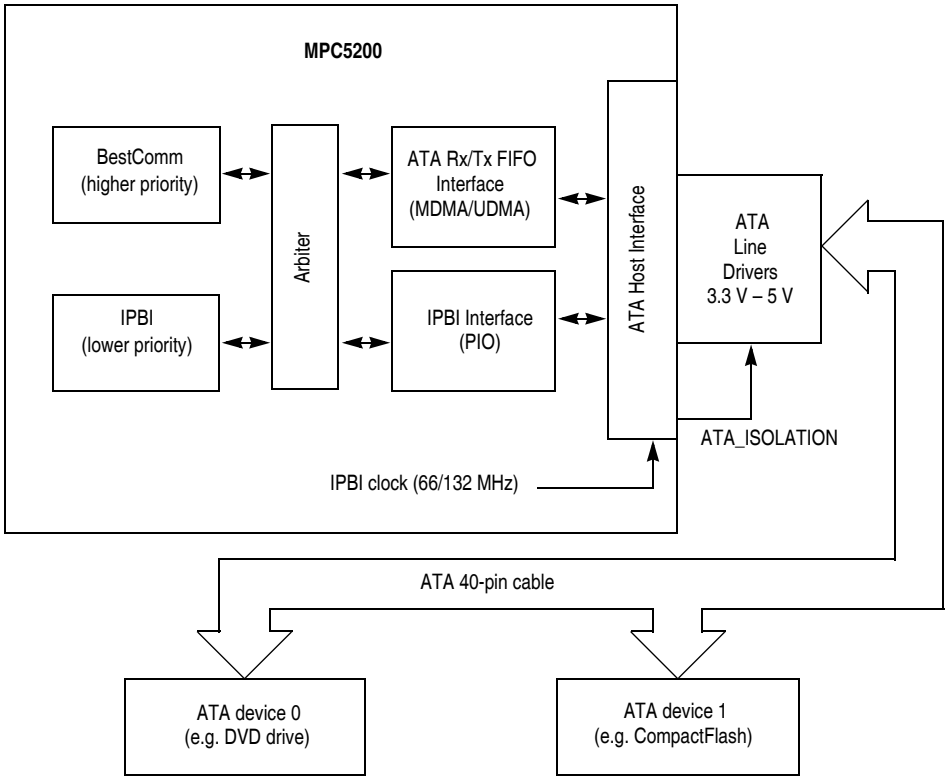

Figure 1. ATA Architecture

2.1 Electrical Characteristics

ATA is a voltage level interface and the table below defines the voltage levels for logical ones and zeroes or input and output values.

Table 1. ATA DC Characteristics

Description		Min	Max
I_{oL}	Driver sink current[1]	4 mA	—
I_{oH}	Driver source current[2]	400 µA	—
V_{iH}	Voltage input high (logical one)	2.0 Vdc	—
V_{iL}	Voltage input low (logical zero)	—	0.8 Vdc
V_{oH}	Voltage output high (I_{oH}=-400 µA, logical one)	2.4 Vdc	—
V_{oL}	Voltage output low (I_{oL}= 12 mA, logical zero)	—	0.5 Vdc

NOTES:
1 I_{oL} for DASP shall be 12 mA minimum to meet legacy timing and signal integrity.
2 IoH value at 400 mA is insufficient in the case of DMARQ that is typically pulled low by a 5.6 kW resistor.

2.2 Signals

Table 2. ATA Signals

ATA Signal	ATA Acronym	MPC5200 Pin	Direction		40-Pin ATA Cable
Cable select	CSEL	—		>	28
Chip select 0	CS0-	ATA_CSB0		>	37
Chip select 1	CS1-	ATA_CSB1		>	38
Data bus bit 0	DD0	ATA_DATA_0	<>		17
Data bus bit 1	DD1	ATA_DATA_1	<>		15
Data bus bit 2	DD2	ATA_DATA_2	<>		13
Data bus bit 3	DD3	ATA_DATA_3	<>		11
Data bus bit 4	DD4	ATA_DATA_4	<>		9
Data bus bit 5	DD5	ATA_DATA_5	<>		7
Data bus bit 6	DD6	ATA_DATA_6	<>		4
Data bus bit 7	DD7	ATA_DATA_7	<>		3
Data bus bit 8	DD8	ATA_DATA_8	<>		4
Data bus bit 9	DD9	ATA_DATA_9	<>		6
Data bus bit 10	DD10	ATA_DATA_10	<>		8
Data bus bit 11	DD11	ATA_DATA_11	<>		10
Data bus bit 12	DD12	ATA_DATA_12	<>		12
Data bus bit 13	DD13	ATA_DATA_13	<>		14

Table 2. ATA Signals (continued)

ATA Signal	ATA Acronym	MPC5200 Pin	Direction		40-Pin ATA Cable
Data bus bit 14	DD13	ATA_DATA_14	<>		16
Data bus bit 15	DD14	ATA_DATA_15	<>		18
Device active or slave (Device 1) present	DASP-	—		>	39
Device address bit 0	DA0	ATA_SA_0		>	35
Device address bit 1	DA1	ATA_SA_1		>	33
Device address bit 2	DA2	ATA_SA_2		>	36
DMA acknowledge	DMACK-	ATA_DACK_B		>	29
DMA request	DMARQ	ATA_DRQ	<		21
Interrupt request	INTRQ	ATA_INTRQ	<		31
I/O read DMA ready during Ultra DMA data in bursts Data strobe during Ultra DMA data out bursts	DIOR- HDMARDY- HSTROBE	ATA_IOR_B		> > >	25
I/O ready DMA ready during Ultra DMA data out bursts Data strobe during Ultra DMA data in bursts	IORDY- DDMARDY- DSTROBE	ATA_IOCHRDY	< < <		27
I/O write Stop during Ultra DMA data bursts	DIOW- STOP	ATA_IOW_B		> >	23
Passed diagnostics Cable assembly type identifier	PDIAG- CBLID-	-	<>		34
Reset	RESET-	HRESET_B	>		1
Ground	GND				2,19,22,24, 26, 30,40

2.3 IO Cable

The cable specification impacts system integrity and the maximum length that shall be supported in any application. Cable total length shall not exceed 0.46 m (18 in.). Cable capacitance shall not exceed 35 pF. 80-pin grounded cable helps to avoid problems with noise and inductance although 40-pin cable is sufficient for modes up to UDMA-2 as in the ATA/ATAPI-4 spec. Also ATA/ATAPI-4 compliant termination is a must; otherwise, CRC checksum error may randomly occur.

2.4 Byte Ordering

Assuming a block of data contains "*n*" bytes of information, the bytes are labeled Byte(0) through Byte(*n*-1), where Byte(0) is the first byte of the block, and Byte(*n*-1) is the last byte of the block. Table 3 shows the order of the bytes on the ATA interface.

Table 3. ATA Byte Ordering

	DD 15	DD 14	DD 13	DD 12	DD 11	DD 10	DD 9	DD 8	DD 7	DD 6	DD 5	DD 4	DD 3	DD 2	DD 1	DD 0
First transfer	Byte (1)								Byte (0)							
Second transfer	Byte (3)								Byte (2)							
...																
Last transfer	Byte (*n*-1)								Byte (*n*-2)							

2.5 Register Addressing

This section describes the generation of the actual address (CS*n*-, DA*n*) that is present on the physical ATA interface to address an ATA drive register by means of operations on internal registers of the MPC5200. Table 4 shows how internal registers are decoded to ATA bus operations.

Table 4. ATA Register Address/Chip Select Decoding

Addresses						Functions	
MPC5200 register address	CS0-	CS1-	DA2	DA1	DA0	READ (DIOR-)	WRITE (DIOW-)
						Control Block Registers	
—	1	1	x	x	x	Data bus high impedance	Not used
—	1	0	0	x	x	Data bus high impedance	Not used
—	1	0	1	0	x	Data bus high impedance	Not used
0x3A5C	1	0	1	1	0	Drive alternate status	Drive device control
—	1	0	1	1	1	Obsolete	Not used
0x3A60	0	1	0	0	0	Drive data	Drive data
0x3A64	0	1	0	0	1	Drive error	Drive features
0x3A68	0	1	0	1	0	Drive sector count	Drive sector count
0x3A6c	0	1	0	1	1	Drive sector number	Drive sector number
						Drive LBA bits 0-7*	Drive LBA bits 0-7*
0x3A70	0	1	1	0	0	Drive cylinder low	Drive cylinder low
						Drive LBA bits 8-15*	Drive LBA bits 8-15*
0x3A74	0	1	1	0	1	Drive cylinder high	Drive cylinder high
						Drive LBA bits 16-23*	Drive LBA bits 16-23*

Table 4. ATA Register Address/Chip Select Decoding (continued)

0x3A78	0	1	1	1	0	Drive device/head	Drive device/head
						Drive LBA bits 24-27*	Drive LBA bits 24-27*
0x3A7C	0	1	1	1	1	Drive device status	Drive command
—	0	0	x	x	x	Invalid address	Invalid address
*Mapping of registers in LBA mode							

2.6 Sector Addressing

The addressing of data sectors recorded on the device's media is performed by a logical sector address. Two modes are supported:

- Cylinder/head/sector addressing (CHS)
- Logical block addressing (LBA)

The MPC5200 host system may select either CHS translation addressing or LBA addressing on a command-by-command basis by using the LBA bit in the ATA drive device/head register. The LBA bit must be set if the host uses LBA addressing mode. The MPC5200 allows 27-bit LBA addressing per device.

2.7 ATA Programming Model

Figure 2 describes the typical structure and sequence of steps needed to initialize and issue ATA commands for MPC5200.

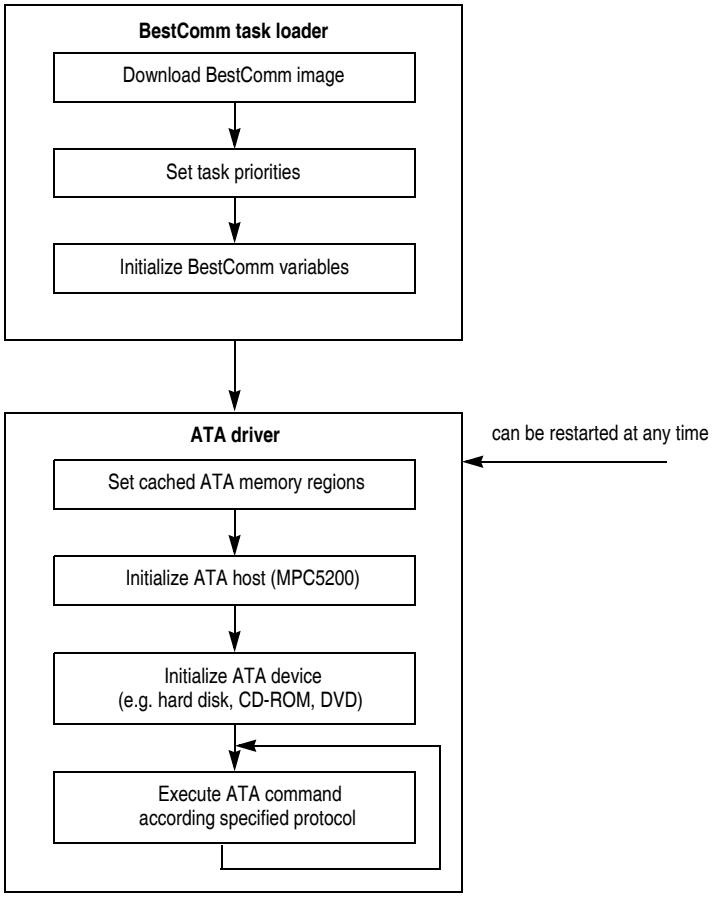

Figure 2. MPC5200 ATA Programming Model

A BestComm task loader is needed if there is some peripheral requiring DMA transfer over the FIFOs to/from a peripheral (ATA MDMA/UDMA, Ethernet, LocalPlus, PCI, etc.). The BestComm image is downloaded and initialized only once, right after booting, due to the static content of the BestComm image (image means BestComm binary firmware). The BestComm image acts as a server that serves clients (peripheral software drivers).

It is recommended to start with the software examples published on the sample code CD. These software examples include a set of already “pre-cooked” BestComm tasks that handle all required peripherals via the BestComm C-API (BAPI). The system integrator needs to recompile all drivers in the case of any change in the image. This guarantees compatible versions for both BestComm firmware and software drivers.

Figure 3 documents the structure for the MDMA/UDMA data transfer.

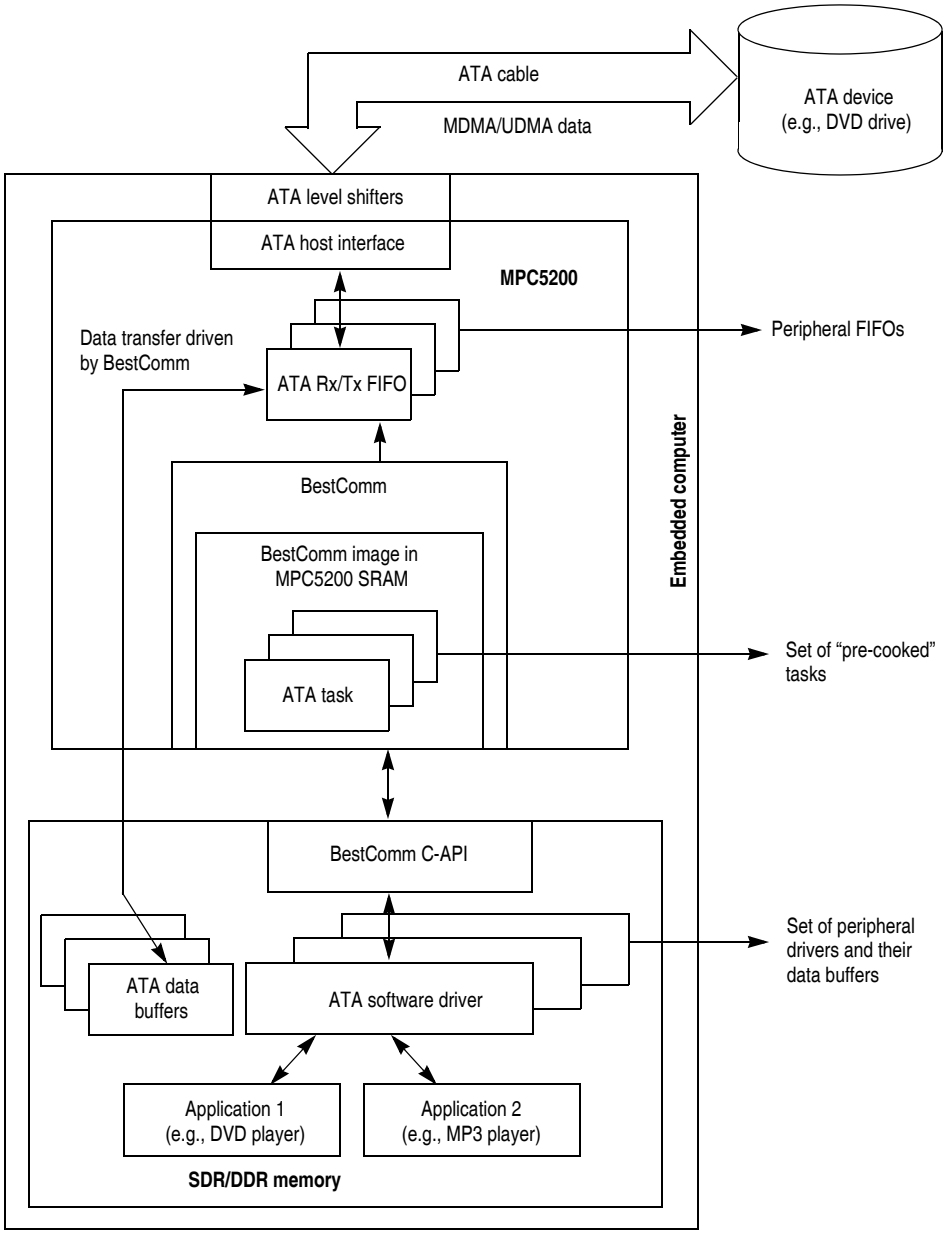

Figure 3. MDMA/UDMA BestComm Driven Data Transfer

The BestComm C-API is not resident software. It is a piece of software statically linked to the driver that includes a set of standardized function calls.

The following steps should be used to set up the cache for ATA memory buffer regions:

- Set DBAT registers and enable data cache.[2] (See *Programming Environments Manual for 32-Bit Implementations of the PowerPC Architecture* for details.)
- Set XLB priorities in the XLB arbiter master priority register and enable XLB priorities in the arbiter master priority enable register.[3] (See the XLB Arbiter chapter in the *MPC5200 User's Manual* for details.)
- Set window address, size and snooping policy in the XLB arbiter snoop window register and enable snooping in the XLB arbiter configuration register.[4] (See the XLB Arbiter chapter in the *MPC5200 User's Manual* for details.)

2.8 Host Initialization

The MPC5200 ATA host initialization covers three different groups of steps:

1. ATA common initialization
   - Set ATA chip select in GPS port configuration register
   - Read IPBI clock speed in CDM configuration register (66 or 132 MHz) needed for timing registers)[5]
   - Enable ATA clock in CDM clock enable register
   - Install ATA interrupt handler and BestComm interrupt handler for MDMA/UDMA modes
   - Enable ATA/BestComm interrupt
2. PIO initialization (BestComm not needed for PIO modes)
   - Set PIO timing registers
   - Enable drive interrupt (to pass to CPU in PIO modes) in ATA host configuration register (IE bit)
   - Enable IORDY in ATA host configuration register (for PIO-3 and above)
3. MDMA/UDMA initialization
   - Set either MDMA or UDMA timing registers

2.9 Protocols

ATA commands written into the ATA drive device command register are grouped into different classes according to the protocols. The protocols define the command execution flow. The command classes with their associated protocols are defined in Table 5.

The following sections directly document the commands instead of providing a detailed general description for each protocol. The addressing scheme and the handling of some of the commands are usually provided in linkable libraries by operating system vendors; therefore, the following description is

2. Usually entire memory cached, initialized by bootloader
3. Initialized by bootloader
4. Set up by driver
5. Bootloader sets up IPBI clock speed, peripheral drivers should read status only

intended to help software engineers understand how to initialize the MPC5200 ATA interface and how to execute standard commands.

NOTE

Flash storage media require a different set of commands that are not described in this application note. It is recommended to first read the identify table for each ATA device found on the ATA bus. It contains specific information needed to handle the device properly. For protocol and identification information details, see *Attachment with Packet Interface Extension,* ANSI NCITS 317-1998.

Table 5. ATA Protocols

Protocol	Commands
Device reset protocol	—
Execute device diagnostic protocol	—
Device selection protocol	—
PIO data-in command protocol	CFA TRANSLATE SECTOR, IDENTIFY DEVICE, IDENTIFY PACKET DEVICE, READ BUFFER, READ MULTIPLE, READ SECTOR(S), SMART READ DATA
PIO data-out command protocol	CFA WRITE MULTIPLE WITHOUT ERASE, CFA WRITE SECTORS WITHOUT ERASE, DOWNLOAD MICROCODE, SECURITY DISABLE PASSWORD, SECURITY ERASE UNIT, SECURITY SET PASSWORD, SECURITY UNLOCK, WRITE BUFFER, WRITE MULTIPLE, WRITE SECTOR(S)
Non-data command protocol	CFA ERASE SECTORS, CFA REQUEST EXTENDED ERROR CODE, CHECK POWER MODE, FLUSH CACHE, GET MEDIA STATUS IDLE, IDLE IMMEDIATE, INITIALIZE DEVICE PARAMETERS, MEDIA EJECT, MEDIA LOCK, MEDIA UNLOCK, NOP, READ NATIVE MAX ADDRESS, READ VERIFY SECTOR(S, SECURITY ERASE PREPARE, SECURITY FREEZE LOCK SEEK, SET FEATURES, SET MAX ADDRESS, SET MULTIPLE MODE SLEEP, SMART DISABLE OPERATION, SMART ENABLE/DISABLE AUTOSAVE, SMART ENABLE OPERATION, SMART EXECUTE OFFLINE IMMEDIATE, SMART RETURN STATUS, STANDBY, STANDBY IMMEDIATE
DMA command protocol	READ DMA, WRITE DMA
Packet non-data and PIO data command protocol	PACKET, SERVICE
Packet DMA command protocol	PACKET, SERVICE
Read/write DMA queued command protocol	READ DMA QUEUED, SERVICE, WRITE DMA QUEUED

2.9.1 Drive Initialization

2.9.1.1 PIO Initialization Protocol Steps

1. Select drive according to device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY[6]=0 and DRQ[7] = 0 in ATA drive alternate status register

6. BSY - The Drive Busy bit indicates that the device is busy.
7. DRQ - The Data Request bit indicates that the device is ready to transfer a word of data between the host and the device.

b) Write ATA drive device/head register with appropriate DEV[8] bit and LBA bit
c) Wait 400 ns
d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register

2. Enable INTRQ in ATA drive device control register (nIEN bit)
3. Set PIO mode in ATA drive features register according to non-data command protocol in ATA/ATAPI-4 spec:
   a) Set PIO mode in ATA drive sector count register
   b) Write SET TRANSFER mode into ATA drive features register
   c) Write SET FEATURES command into ATA drive device command register
   d) Wait for ATA interrupt – indicates that the device has accepted the SET FEATURES command

2.9.1.2 MDMA/UDMA Initialization Protocol Steps

1. Select drive according to device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
   b) Write ATA drive device/head register with appropriate DEV bit and LBA bit
   c) Wait 400 ns
   d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
2. Enable INTRQ in ATA drive device control register (nIEN bit)
3. Set MDMA/UDMA mode in ATA drive features register according to non-data command protocol in ATA/ATAPI-4 spec:
   a) Set MDMA/UDMA mode in ATA drive sector count register
   b) Write SET TRANSFER MODE into ATA drive features register
   c) Write SET FEATURES command into ATA drive device command register
   d) Wait for ATA interrupt – indicates that the device has accepted the SET FEATURES command

8. DEV - The Device Selection bit in ATA drive device/head register specifies which device is selected, 0 or 1.

2.9.2 PIO Data Read

2.9.2.1 PIO Data in Command Protocol Steps

1. Select drive according to device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
   b) Write ATA drive device/head register with appropriate DEV bit and LBA bit
   c) Wait 400 ns
   d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
2. Write ATA drive cylinder low register, ATA drive cylinder high register, ATA drive sector number register, ATA drive sector count register
3. Write READ SECTOR(S) command into ATA drive device command register
4. Wait 400 ns
5. Wait for ATA interrupt – indicates that the data is ready for reading by host
6. Read (sector_count × sector_size) 16-bit ATA data words from ATA drive data register driven either by core or Bestcomm. In the case of Bestcomm when snooping is not used, cache needs to be invalidated first

2.9.3 PIO Data Write

2.9.3.1 PIO Data Out Command Protocol Steps

1. Select drive according to device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
   b) Write ATA drive device/head register with appropriate DEV bit and LBA bit
   c) Wait 400 ns
   d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
2. Write ATA drive cylinder low register, ATA drive cylinder high register, ATA drive sector number register, ATA drive sector count register
3. Write WRITE SECTOR(S) command into ATA drive device command register
4. Wait 400 ns
5. Wait for BSY=0 and DRQ = 1 in ATA drive alternate status register
6. Write (sector_count × sector_size) 16-bit ATA data words to ATA drive data register, driven either by core or Bestcomm. In the case of Bestcomm when snooping is not used, cache needs to be flushed first (entire write buffer)
7. Wait for ATA interrupt – indicates that the data has been written by the device

2.9.4 DMA Data Read

2.9.4.1 DMA Command Protocol Steps

1. Select drive according to the device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
   b) Write ATA drive device/head register with appropriate DEV bit and LBA bit
   c) Wait 400 ns
   d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
2. If snooping is not used, cache needs to be invalidated
3. Initialize and start BestComm ATA task
4. Set ATA FIFO alarm and granularity in ATA Rx/Tx FIFO alarm register, ATA Rx/Tx FIFO control register
5. Set FIFO Reset bit (FR) in ATA drive device command register when the direction is switched from Tx to Rx
6. Clear FIFO Reset bit (FR) in ATA drive device command register*
7. Set FIFO Flush in Rx Mode bit (FE) and Read bit (READ) in ATA drive device command register*
8. If UDMA, set UDMA bit (UDAMA) in ATA drive device command register. Clear it for MDMA*
9. Set Drive Interrupt bit (IE) in ATA drive device command register*

NOTE

This could be done just by one command.

10. Wait 400 ns
11. Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
12. Write ATA drive cylinder low register, ATA drive cylinder high register, ATA drive sector number register, ATA drive sector count register
13. Wait 400 ns
14. Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
15. Write READ DMA command into ATA drive device command register
16. Wait for BestComm interrupt – indicates all data moved from ATA Rx FIFO to the RAM

2.9.5 DMA Data Write

2.9.5.1 DMA Command Protocol Steps

1. Select drive according to device selection protocol in ATA/ATAPI-4 spec:
   a) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
   b) Write ATA drive device/head register with appropriate DEV bit and LBA bit
   c) Wait 400 ns
   d) Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register

2. If snooping is not used, cache needs to be flushed (entire write buffer)
3. Initialize and start BestComm ATA task
4. Set ATA FIFO alarm and granularity in ATA Rx/Tx FIFO alarm register, ATA Rx/Tx FIFO control register
5. Set Write bit (WRITE) in ATA drive device command register*
6. If UDMA, set UDMA bit (UDAMA) in ATA drive device command register. Clear it for MDMA*
7. Set Drive Interrupt (IE) bit in ATA drive device command register*

NOTE

This could be done just by one command.

8. Wait 400ns
9. Wait for BSY=0 and DRQ = 0 in ATA drive alternate status register
10. Write ATA drive cylinder low register, ATA drive cylinder high register, ATA drive sector number register, ATA drive sector count register
11. Write WRITE DMA command into ATA drive device command register
12. Wait for ATA interrupt – indicates that the data has been written by the device

2.9.6 Interrupt Handler

In the interrupt service routine, the first action is to clear the lower part of the ATA drive device command register by writing a logical zero (bits HUT, FR, FE, IE, UDAMA, READ, WRITE). The pending ATA drive interrupt is cleared by reading of the ATA drive device sstatus register as it is specified in the ATA/ATAPI-4 specification.

3 Performance Analysis

Table 6 documents the maximum data throughput for all possible ATA modes. Of course, it is a theoretical maximum throughput that, in reality, is never achieved. The real final result depends on the following:

- *Device response delay* – Average seek time for hard disks is a few milliseconds; therefore, some host-based cache would be essential for applications like video players where large and continuous streams of data need to be read from the device. This kind of host-based cache is usually implemented as filesystem cache
- *Sustained data rate* – Devices like hard disks have limited speed at which physical data is read. This number can be found in the device data sheet.
- *Mode dependent delay*
  - Standby mode – The ATA device goes to the standby mode when the standby timer expires. The interface is capable of accepting commands while waiting for the spindle to reach operating speed. Delay is roughly hundreds of milliseconds
  - Idle mode – See technical documentation for the ATA device.

- *Device fragmentation delay* – If the device is highly defragmented, using the look-ahead feature[9] does not help. This is a typical problem of hard disks or Flash storage devices. DVDs and CDs always have contiguous blocks of data.
- *MPC5200 timing granularity* – The MPC5200 ATA interface is clocked by the IPBI clock (66/132 MHz). The ATA/ATAPI-4 specification defines minimum values for cycle time, setup time, and hold time. Minimum values for timing registers are specified in IPBI clock cycles and need to be rounded up; therefore, the best result are achieved with finer granularity. (For example, the IPBI clock 132 MHz is measured on the logic analyzer, 31.5 MBytes/sec, for UDMA-2.) Compare this with the maximum theoretical throughput in Table 6.

Table 6. ATA Maximum Data Throughput

Mode	Max.Rate [MBytes/Sec]	MPC5200 ATA Initialization Driven By	MPC5200 ATA Data Transfer Driven By
PIO-0	3.3	core	core/Bestcomm
PIO-1	5.2	core	core/Bestcomm
PIO-2	8.3	core	core/Bestcomm
PIO-3	11.1	core	core/Bestcomm
PIO-4	16.7	core	core/Bestcomm
MDMA-0	4.2	core	BestComm
MDMA-1	13.3	core	BestComm
MDMA-2	16.7	core	BestComm
UDMA-0	16.7	core	BestComm
UDMA-1	25	core	BestComm
UDMA-2	33	core	BestComm

ATA performance can be optimized by:

- Reading/writing of the maximum number of sectors per single ATA command (see Figure 4[10]). Regular hard disk defragmentation is essential
- Enabling of special features in the ATA device—look-ahead, write cache, etc. First, they need to be checked in the identify table to see if supported and then enabled by the SET FEATURES command
- Using IPBI 132 MHz as described above
- Utilizing Bestcomm which can drive read/write in PIO mode to offload the core

9. Look-ahead read allows reading consequent sectors into the device cache. ATA host reads data from device cache. The size of cache is vendor/type dependent, which can be found in the device documentation

10.Normalized results are measured by writing/reading 1 MB data to/from the same LBA address for IBM DBCA-203240 hard disk with disabled drive cache. It could be different for other ATA devices. Check maximum number of sectors per single command in the device identify table (usually up to 16 sectors per single command supported).

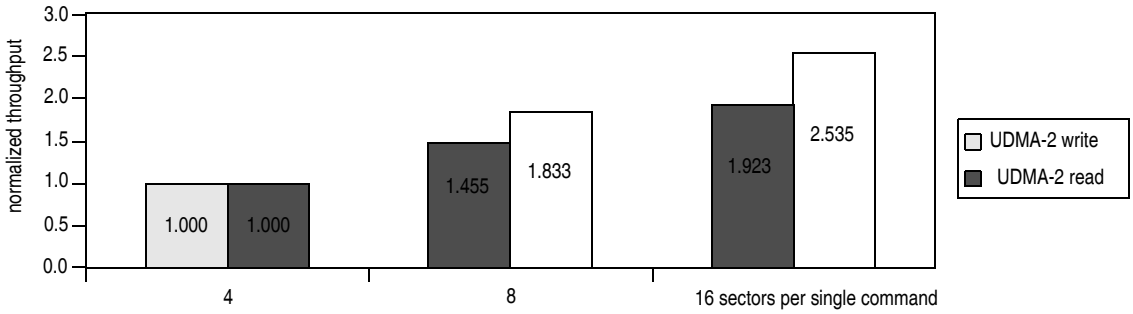

Figure 4. ATA Relative Throughput Improvement

System throughput is a result of the speed of the peripheral and the final integration—typically in OSs. For a properly defragmented ATA device, it is considered bad programming style if the MPC5200 ATA host reads or writes only a small number of sectors per single ATA command. In this case, the ATA software driver and its interrupt handler could consume significant amounts of core time. The ATA software driver can block anything running with lower priority in the system (see Table 7).[11] In some cases in OSs, such bad application implementation can degrade final ATA throughput by ~50% .

Table 7. System Throughput

ATA Mode	Sectors Per Single Command	Time Between Two ATA Interrupts
UDMA-2	1	15 µs
	4	60 µs
	16	240 µs

Looking at Table 7, the question arises as to why the throughput improvement indicated in Figure 4 is not as expected. The answer is simple; it is writing/reading the same LBA address with the look-ahead/write cache feature disabled. The time needed to read data from the interface is much smaller (microseconds) than the average seek time (milliseconds).

Static memory allocation for the ATA software driver is recommended.

Although not described in the *MPC5200 User's Manual*, the MPC5200 ATA interface is capable of UDMA-4 66 MBytes/sec mode. The programming model is the same as for UDMA-2. The impact of all the above possible performance considerations is more visible because of higher speed.

11.Reading subsequent sectors from hard disk with enabled look-ahead feature, the worst scenario for MPC5200 core load

4 Literature

- MPC5200UM, *MPC5200 User's Manual* (www.freescale.com)
- MPCFPE32B, *Programming Environments Manual for 32-Bit Implementations of the PowerPC Architecture*) (www.freescale.com)
- G2CORERM, *G2 Core Reference Manual* (www.freescale.com)
- ATA/ATAPI-4, *AT Attachment with Packet Interface Extension*, ANSI NCITS 317–1998 (www.t13.org/technical/d98121r0.pdf)
- G2CORERM, *G2 PowerPC Core Reference Manual* (www.freescale.com)

Appendix A MPC5200 ATA Register Summary

Table A1. MPC5200 ATA Host Registers

Register	Acronym[1]	Offset	Size [Bits]
ATA host configuration	ata_config	MBAR+0x3A00	8
ATA host status	ata_status	MBAR+0x3A04	8

NOTES:
1 Acronyms use terminology such as ata.h in freely distributed software examples.

Table A2. MPC5200 ATA Timing Registers

Register	Acronym[1]	Offset	Size [Bits]
ATA PIO timing 1	ata_pio1	MBAR+0x3A08	32
ATA PIO timing 2	ata_pio2	MBAR+0x3A0C	32
ATA multiword DMA timing 1	ata_dma1	MBAR+0x3A10	32
ATA multiword DMA timing 2	ata_dma2	MBAR+0x3A14	32
ATA ultra DMA timing 1	ata_udma1	MBAR+0x3A18	32
ATA ultra DMA timing 2	ata_udma2	MBAR+0x3A1C	32
ATA ultra DMA timing 3	ata_udma3	MBAR+0x3A20	32
ATA ultra DMA timing 4	ata_udma4	MBAR+0x3A24	32
ATA ultra DMA timing 5	ata_udma5	MBAR+0x3A28	32
ATA share count	ata_invalid	MBAR+0x3A2C	32

NOTES:
1 Acronyms use terminology such as ata.h in freely distributed software examples.

Table A3. MPC5200 ATA FIFO Registers

Register	Acronym[1]	Offset	Size [Bits]
ATA Rx/Tx FIFO data word	ata_fifo_data	MBAR+0x3A3C	32
ATA Rx/Tx FIFO status	ata_fifo_status	MBAR+0x3A41	8
ATA Rx/Tx FIFO control	ata_fifo_control	MBAR+0x3A44	8
ATA Rx/Tx FIFO alarm	ata_fifo_alarm	MBAR+0x3A4A	16
ATA Rx/Tx FIFO read pointer	ata_fifo_rdp	MBAR+0x3A4E	16
ATA Rx/Tx FIFO write pointer	ata_fifo_wrp	MBAR+0x3A52	16

NOTES:
1 Acronyms use terminology such as ata.h in freely distributed software examples.

Table A4. MPC5200 ATA Drive Registers

Register	Acronym[1]	Offset	Size [Bits]	Mode
ATA drive device control	ata_drive_ctrl	MBAR+0x3A5C	8	write-only
ATA drive alternate status			8	read-only
ATA drive data	ata_drive_data	MBAR+0x3A60	16	R/W
ATA drive features	ata_drive_ftr	MBAR+0x3A64	8	write-only
ATA drive error			8	read-only
ATA drive sector count	ata_drive_sc	MBAR+0x3A68	8	R/W
ATA drive sector number	ata_drive_sn	MBAR+0x3A6C	8	R/W
ATA drive cylinder low	ata_drive_cl	MBAR+0x3A70	8	R/W
ATA drive cylinder high	ata_drive_ch	MBAR+0x3A74	8	R/W
ATA drive device/head	ata_drive_dh	MBAR+0x3A78	8	R/W
ATA drive device command	ata_drive_cmd	MBAR+0x3A7C	8	write-only
ATA drive device status			8	read-only

NOTES:
1 Acronyms use terminology such as ata.h in freely distributed software examples.

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.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
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

AN3223
Rev. 0, 2/2006

freescale™
semiconductor