MP3 Personal Digital Players Using Actel FPGAs

Introduction

The Moving Pictures Expert Group (MPEG) was formed in 1988 to settle on a single codec (compression/decompression) scheme for digital audio. By 1992, the International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) had created a standard for audio and video coding called MPEG1 (ISO/IEC 11172). This enabled the industry to agree on a single format to handle all the various types of audio/visual media that were emerging in the new digital age.

The Personal Digital Player market is beginning to gain tremendous momentum with the development of the powerful MPEG Audio Layer 3 (MP3) format used for storing and playing back music in digital form. An MP3 digital file of an uncompressed audio track is only about one-tenth the size of the original track. A standard Read/Write CD that holds 650 Mbytes at 74 minutes of digital audio recording time can hold about 740 minutes of MP3 compressed audio, which is a little more than 12 hours per CD.

Typical solutions for the Digital Player are available in software form as "Internet Audio Players," based on Windows or other OS on the PC, or in hardware form such as "portable MP3 players," which resemble a Walkman or a pager. The hardware players were pioneered by Diamond Multimedia in the form of their Rio PMP300 and Rio PMP500 MP3 Digital Players. These players used Actel's A40MX04 and A42MX09 FPGAs respectively to implement application-specific functions. In each case, these Actel FPGAs met the price, power, and performance requirements of the overall systems.

The A40MX04 FPGA used in the Rio PMP300 provides 547 logic modules, which can be configured as combinatorial logic or up to 273 flip-flops and 69 I/Os for implementing the interface logic. Power consumption is critical in any portable product and the A40MX04 meets the 10mW maximum power requirements of the Diamond MP3 player. Because of its antifuse switch technology, the one-time programmable A40MX04 FPGA burns less power compared to its reprogrammable counterparts with their SRAM switch technology. Therefore, it is more suited to this application space, consuming only 5% of the Rio PMP300 total power consumption. The A40MX04 and A42MX09 FPGAs easily meet the 48 MHz speed requirements to provide a highly cost-effective solution. The flexible architecture of Actel's MX devices enabled the system designer to optimally implement the various functions in each MP3 player.

The Rio PMP300 MP3 Digital Player is described in this application note. The role of Actel's A40MX04 FPGA in the implementation is discussed, showing how the part met the functional, speed, power, and system-level requirements in a cost-effective manner. Additionally, an overview of the use of Actel's A42MX09 in the PMP500 is given.

MP3 Digital Player System Requirements

The Rio PMP300 is a portable MP3 music player that stores up to 60 minutes of MP3 digital quality audio data with 32 MB of built-in Flash memory. Playback time can be increased by removing the Flash memory card and replacing it with a higher-capacity memory card.

Music is downloaded to the player through a parallel port at a transfer rate of approximately 10 seconds per 1 MB of encoded music, taking about 6 minutes to fill up a 32 MB Flash card.

The player comes with software that can be loaded on a PC. The software provides the user interface to select, sequence, and download MP3 files of various songs to the player through the PC's parallel port. The software can also convert music CDs to MP3 format files that can then be downloaded to the player. Figure 1 on page 2 shows the system-level block diagram.

The MP3 player has two basic modes of operation—download mode and playback mode. In download mode, the software application running on the PC will allow the user to download MP3 files into the Flash memory in the player. Once all the files have been downloaded, the player can be disconnected from the PC and carried around to listen to the downloaded music via the headphones. The Rio PMP300 has controls for the various playback options such as play, fast-forward, fast-rewind, stop/pause, volume control, random play, and repeat play functions.

The small size of the MP3 player places some significant constraints on the physical size of the components used in the design. In addition, the MP3 player is a portable, battery powered appliance that needs to operate with a very low power consumption of about 180mW. Both of these factors strongly influence the choice of components for the functional design.

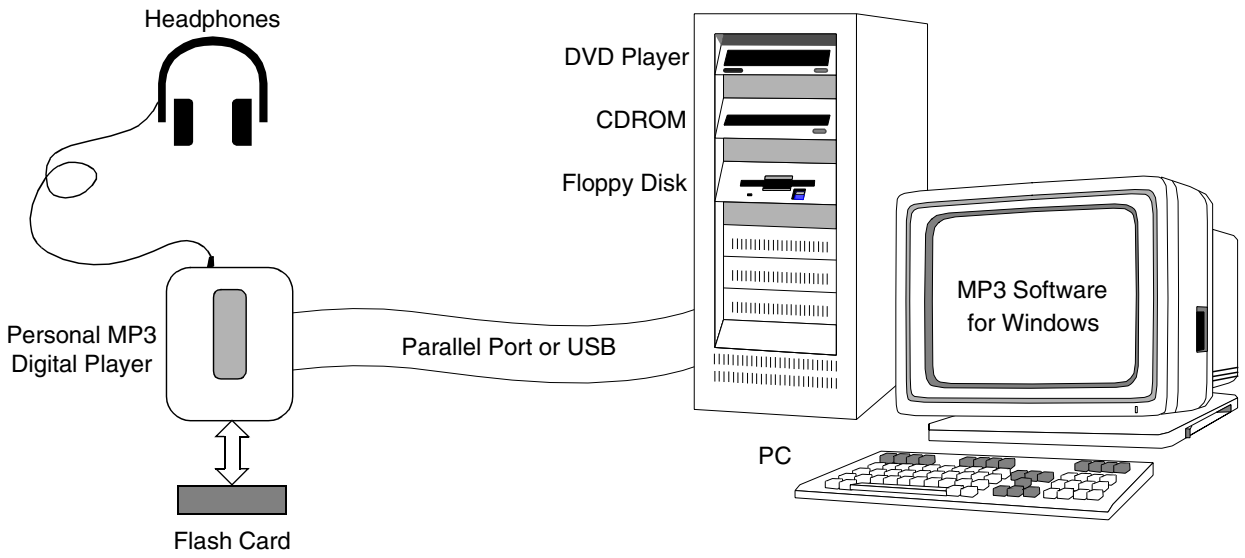


Figure 1 • *System-Level Block Diagram*

MX Architectural Highlights and Benefits

Logic Modules

The 40MX and 42MX Actel FPGA devices are composed of fine-grained building blocks that enable fast, efficient logic designs. These devices consist of logic modules, routing resources, clock networks, and I/O modules.

The 40MX logic module is a 4-input, 1-output logic circuit designed to implement a wide range of logic functions with efficient use of interconnect routing resources (Figure 2). The logic module can implement the four basic logic functions (NAND, AND, OR and NOR) in gates of two, three or four inputs, as well as a a variety of D-latches, exclusivity functions, AND-ORs, and OR-ANDs. No dedicated hardwired latches or flip-flops are required in the array, since latches and flip-flops can be constructed from the logic modules when needed.

This architecture gives the designer more flexibility in the utilization of the logic modules, since they can be configured as either combinatorial or sequential circuits. The fine-grained architecture also helps logic partitioning for efficient use of available logic modules. The Rio PMP300 design used only 50 percent of the A40MX04 modules for the entire implementation.

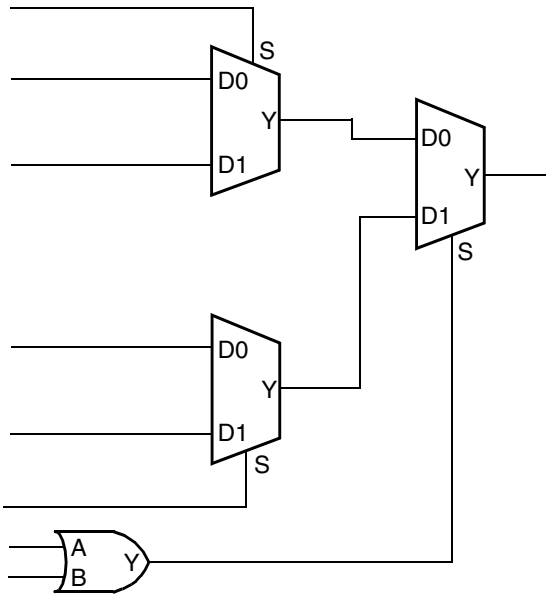


Figure 2 • *40MX Logic Module*

The A42MX09 device was used in the PMP500 model. The 42MX family contains two types of logic modules: combinatorial (C-modules) and sequential (S-modules). The C-module is a 4-input mux with its selects being derived from 4 inputs (Figure 3). This architecture also lends itself to implementations with high utilization. Because the same C-module logic is used as the front-end to the flip-flop for the S-module, registering of a combinatorial function without any additional design changes is easy. In addition, the S-module register can be bypassed so that it implements purely combinatorial logic, adding more flexibility and increasing utilization of logic modules for a given design implementation (Figure 4).

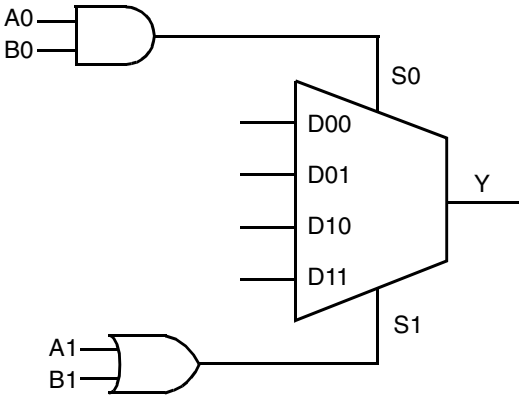


Figure 3 • 42MX C-Module Implementation

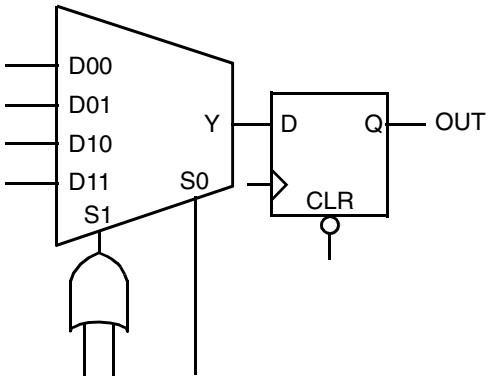


Figure 4 • 42MX S-Module Implementation

I/O Structures

The 42MX devices have configurable I/O modules, giving the user a variety of choices based on design needs (Figure 5).

The I/O modules provide the interface between the device pins and the logic array. All 42MX I/O modules contain a tristate buffer, with input and output latches that can be configured for input, output, or bidirectional operation. Each output pin has a dedicated output-enable control. The I/O module can be used to latch input or output data, or both, providing a fast setup time.

The tristate buffers enable the easy interfacing of MX devices to other busses (such as a system databus, which is usually tristatable) without the need for external devices. In the Rio PMP500, the A42MX09's input latch capability facilitated the latching of address, control, and data signals from the CPU.

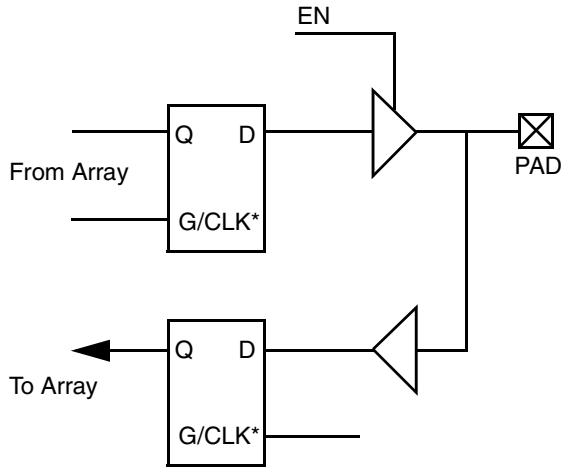


Figure 5 • 42MX I/O Module

Routing Structures

The MX routing structure uses vertical and horizontal routing tracks to interconnect the various logic and I/O modules (Figure 6 on page 4). These routing tracks are metal interconnects that may be of continuous lengths or broken into pieces called segments. Varying segment lengths allows the interconnect of over 90% of design tracks to occur with only two antifuse connections. This helps minimize routing delays between modules, which in turn reduces the path delay from flip-flop to flip-flop (internal paths), enabling higher-frequency designs. Segments can be joined together at the ends using antifuses to increase their lengths up to the full length of the track. All interconnects can be accomplished with a maximum of four antifuses. This again reduces the overall path delays in the chip, resulting

in higher performance. The Rio PMP300 design easily met speed requirements with Actel's 100% automatic place-and-route software. Also, segmented routing promotes low power consumption because each logic module has to drive a minimum of resistive and capacitive loading.

Small Package

Actel's devices use small form factor VQFP packages that consume minimum board space. The A40MX04 is available in an 80-pin VQFP package with a body size of 14mm x 14mm x 1.0mm. The low power dissipation of Actel's FPGAs permits use of this package.

Single Chip

Actel's FPGAs also provide a single-chip solution in contrast to SRAM FPGAs, which require a space-wasting external EPROM device to store the FPGA design. An alternative to this is to store the design file in microprocessor code space, but this requires an additional processor flash and increases the complexity of the processor code. Actel's live-on-power up FPGAs avoid this complication.

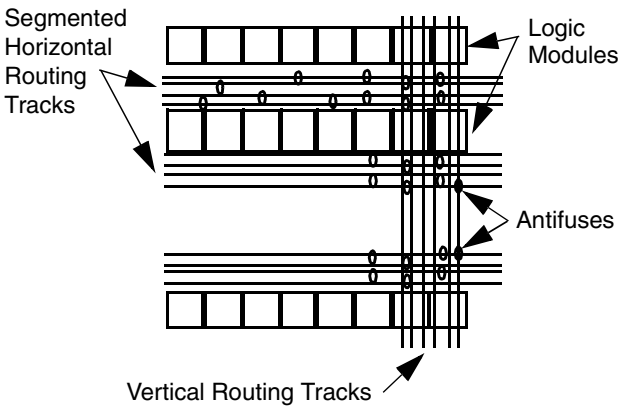


Figure 6 • *MX Routing Structure*

MP3 System Implementation Using A40MX04

Figure 7 shows specific implementation on the Rio PMP300 Digital Player using Actel's A40MX04 FPGA.

Rio PMP300

Parallel I/F Block

The parallel I/F block is used to download data through the parallel port from the PC to the player. Centronix, the parallel port protocol, is the most widely used parallel I/F and the IEEE 1284 parallel port standard provides a compatibility mode to support it. Other MP3 player manufacturers may use other modes, such as enhanced capability port (ECP), to provide faster download capability, which the MX family of devices can easily handle in terms of speed and other requirements.

The Standard Parallel Port (SPP) connector (DB25 or Centronix-36 pin) interface on the PC uses 8 pins for data and up to 9 pins for control/status. Of the 9 control/status pins, 5 are status pins from the peripheral device into the host and 4 are control pins from the host to the peripheral device. These 9 pins are used in different ways, depending on the port's mode of operation.

Figure 8 and Figure 9 on page 5 illustrate the IEEE 1284-compatible interface protocol.

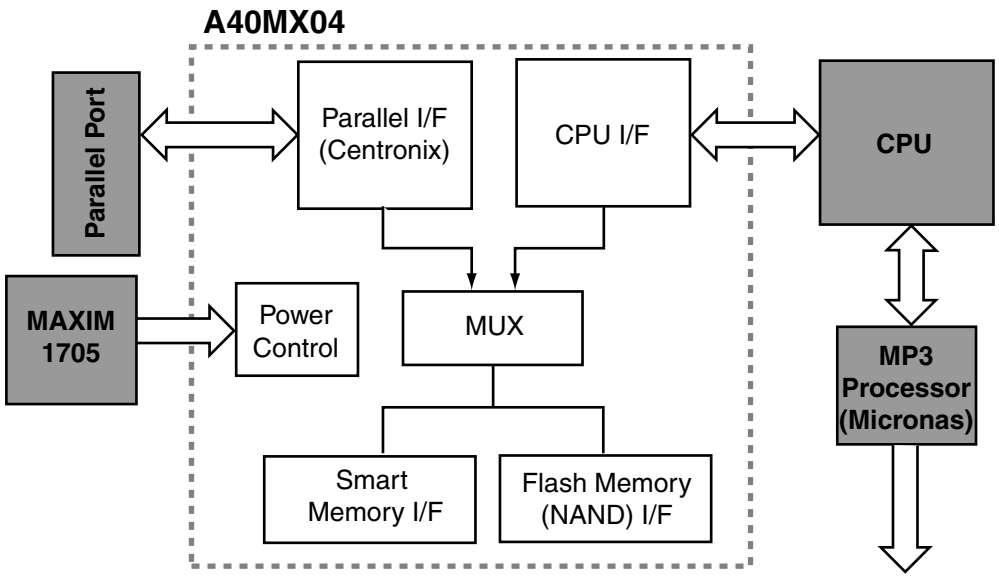


Figure 7 • *Rio PMP300 Block Diagram*

As shown in Figure 1 on page 2, the parallel port in the PC acts as the host device and the FPGA parallel port interface logic acts as the peripheral. The host device places digital data on the databus lines, polls that the BUSY signal from the peripheral device is not asserted, and then asserts the nSTROBE signal. The peripheral device latches the data from the databus and acknowledges the receipt of data by asserting the nACK signal, while the host device deasserts the nSTROBE signal. This sequence repeats until all the required data is transferred to the peripheral device. This is how MP3 digital data is transferred to the MP3 player using the parallel IEEE 1284-compatible mode of operation. Additional control signals, nSELECT-IN and INIT, are used to indicate to the FPGA logic that the parallel port is in download mode and to mark the start of each download operation.

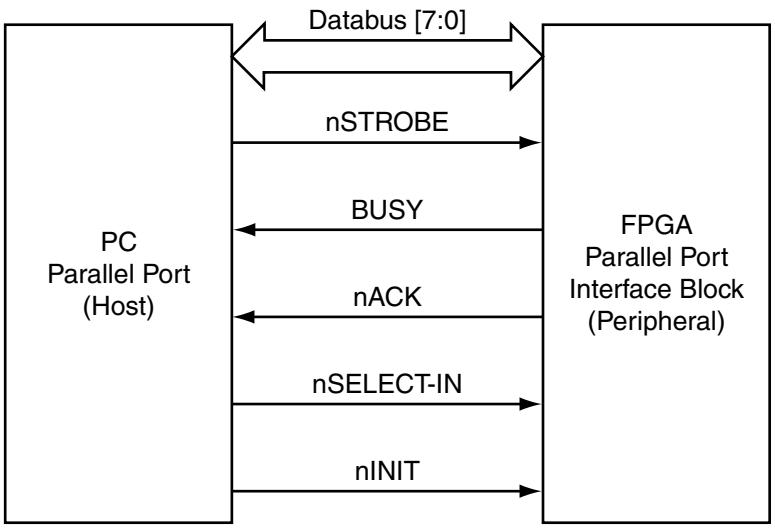


Figure 8 • *Parallel Port Interface Signals*

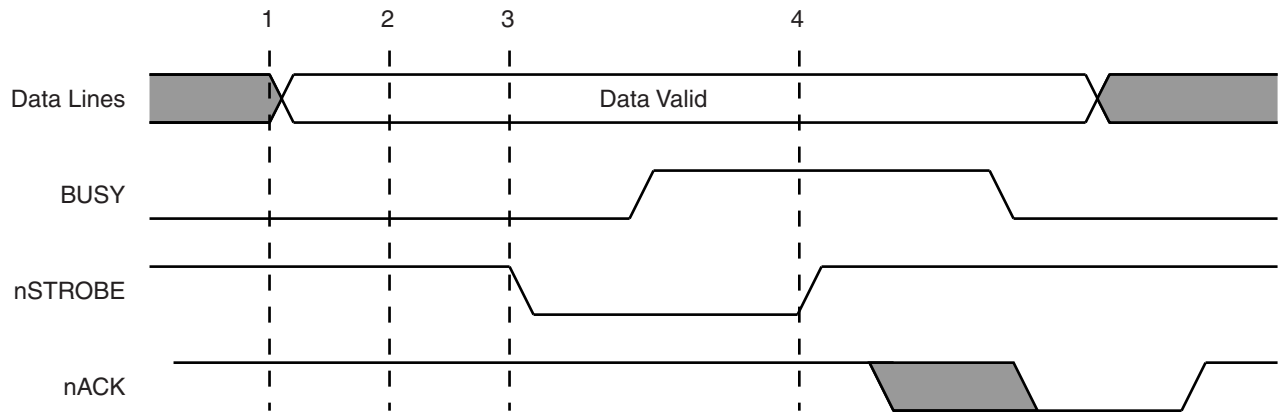


Figure 9 • *Compatibility Mode Data Transfer Cycle*

The FPGA logic needed to accomplish the above operation can be designed as asynchronous logic. This portion of the design used about 100 logic modules in the A40MX04 FPGA device. Speeds of up to 200 Kbytes/second were validated by this design implementation. The device architecture easily accommodated the logic implementation at the required performance level in the lowest speed grade device.

Typical transfer rates in this mode are up to 150 Kbytes/second. By dedicating hardware to toggle these signals on the host and the peripheral device, transfer rates can be increased up to 500 Kbytes/second.

In the ECP mode of data transfer (proposed by HP and Microsoft), high-speed bidirectional communication can be accomplished between the host and peripheral devices. Dedicated signal lines to show the direction of transfer are used along with either data or commands being transferred over the 8 data lines. For more information, refer to the IEEE 1284 specification.

Power Control

Power for the Rio MP3 player is controlled by a MAXIM 1705 device, which is controlled by the Actel FPGA.

If the Rio player is inactive after downloading or playing, the power mode of the MAXIM device is changed appropriately to increase the power efficiency to prolong battery life. It is critical to have this feature for MP3 players, which are typically powered by a single 1.5V battery. The battery should last a minimum of 12 hours to allow you to listen to all of the downloaded music. This power control is one of the salient features of Rio compared with other MP3 players.

The power control device has two power modes of operation. One is a PFM (Pulse-Frequency-Modulation) mode, which is a low quiescent current standby mode that offers a total output current up to 120 mA and reduces quiescent power consumption to 500µW. The other power mode is a PWM (Pulse-Width-Modulation) high-power consumption mode that provides up to 450mA output current. When the MP3 player is in an idle state, the CPU is programmed to write to a register in the FPGA, which generates the MODE signal to put the MAXIM device into the PFM low-power mode. In the PFM mode, Rio PMP300 power dissipation is only 2mW to 3mW. When the MP3 player is active, the MODE signal changes the MAXIM device to the PWM mode. Total power in the active state is around 180mW for the Rio PMP300.

Other FPGA Power Control Issues

Although an external system clock of 14.725 MHz is used, it is divided down in the Actel FPGA to lower frequencies to power up various parts of the system, such as 2.5 MHz for the processor and 32 KHz for the watchdog timer functions. Decreasing the internal switching frequencies helps reduce the overall power in the system since the power dissipation is a function of the switching frequency.

Actel's FPGAs also boast low power consumption. In addition to the inherent low power of Actel's devices due to the antifuse technology, clocks to the various synchronous logic can be shut off when required to further decrease the power consumption that is critical in MP3 player applications. The power due to standby current is typically a small component of the overall power budget. Worst-case standby power for the A40MX04 device is calculated at a very low 6.6mW based on a V_{CC} of 3.3V at an I_{CC} of 2mA.

The active power dissipation is determined by Equation 1:

$$\text{Power } (\mu W) = Ceq * V_{CC}^2 * F \qquad (1)$$

where:

Ceq = equivalent capacitance in pF

V_{CC} = power supply in volts (V)

F = switching frequency in MHz

The components of this power come from the logic modules switching at frequency F, the input and output buffers switching at frequency F, and the power due to the loads on each routed array clock and their equivalent capacitances. The effective loads for the power calculations are smaller, compared to CPLDs or SRAM-based FPGAs, due to the combined effects of shorter segmented routing track lengths and the lower effective RC value for the antifuses that connect these tracks. Active power of A40MX04 in the Rio PMP300 system was measured at only 10mW.

Flash Memory I/F Block

This block is used to control the Flash Memory, which holds the MP3 files that are downloaded through the parallel port. The Flash memory from Samsung provides up to 8M x 8 locations of memory in addition to a spare memory area of 256K x 8 locations. Once the MP3 data is downloaded through the parallel port and written into the Flash memory, the system CPU accesses this data from the Flash memory and writes it to the Micronas MP3 decoder IC. The output of the decoder drives an audio DAC IC, which converts the data to an analog signal to transmit though the headphones.

The interface signals between the Flash memory and the Flash Memory Interface block are shown in Figure 10.

The Flash memory interface controlled by the MX device comprises a multiplexed address/data bus that is 8 bits wide, a Chip Select (CEn), Command Latch Enable (CLEn), Address Latch Enable (ALE), Read Enable (REn), Write Enable (WEn), Write Protect (WPn), Spare Area Enable (SEn), and Ready/Busyn signals.

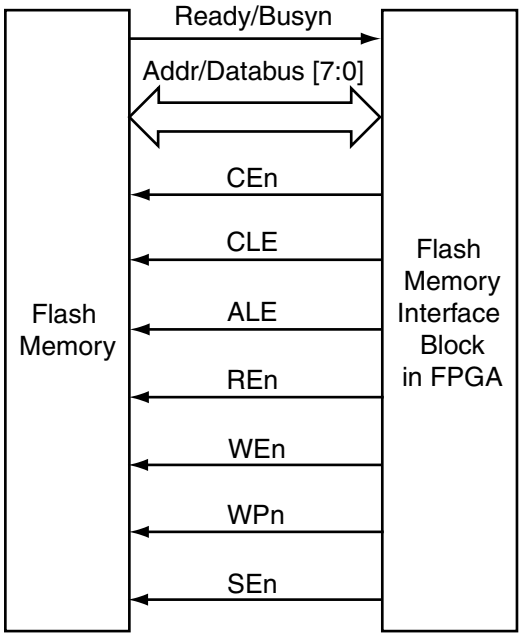


Figure 10 • *Bus Interface*

The Flash memory modes of operation are determined by the following command sets: Sequential Data Input, Read 1, Read 2, Read ID, Reset, Page Program, Block Erase, and Read Status. Each of these commands can be written to the Flash memory from the MX FPGA by asserting the signals required for that particular command operation. Once the command is written to the Flash memory, it is followed by the actual operation of the command such as a Read 1, Program, or Erase. Each of these commands requires an address on the address/data bus followed by either writing or reading the data to/from the specified address, respectively. The various functional waveforms for these commands are in shown Figures 11, 12, and 13.

Commands are written to the Flash memory by asserting CLEn, CEn, and WEn LOW while ALE is held LOW (the command being placed on the address/data bus). All commands require one bus cycle except for the Block Erase command, which requires a two-cycle bus operation.

All operations that require a starting address to be latched by the Flash memory use three bus cycles of addresses: A0~A7, A9~A16, and A17~A22. ALE is driven HIGH to indicate the address latch cycle, WEn is pulsed three times, and the appropriate address is latched on the rising edges of WEn, as shown in Figure 11.

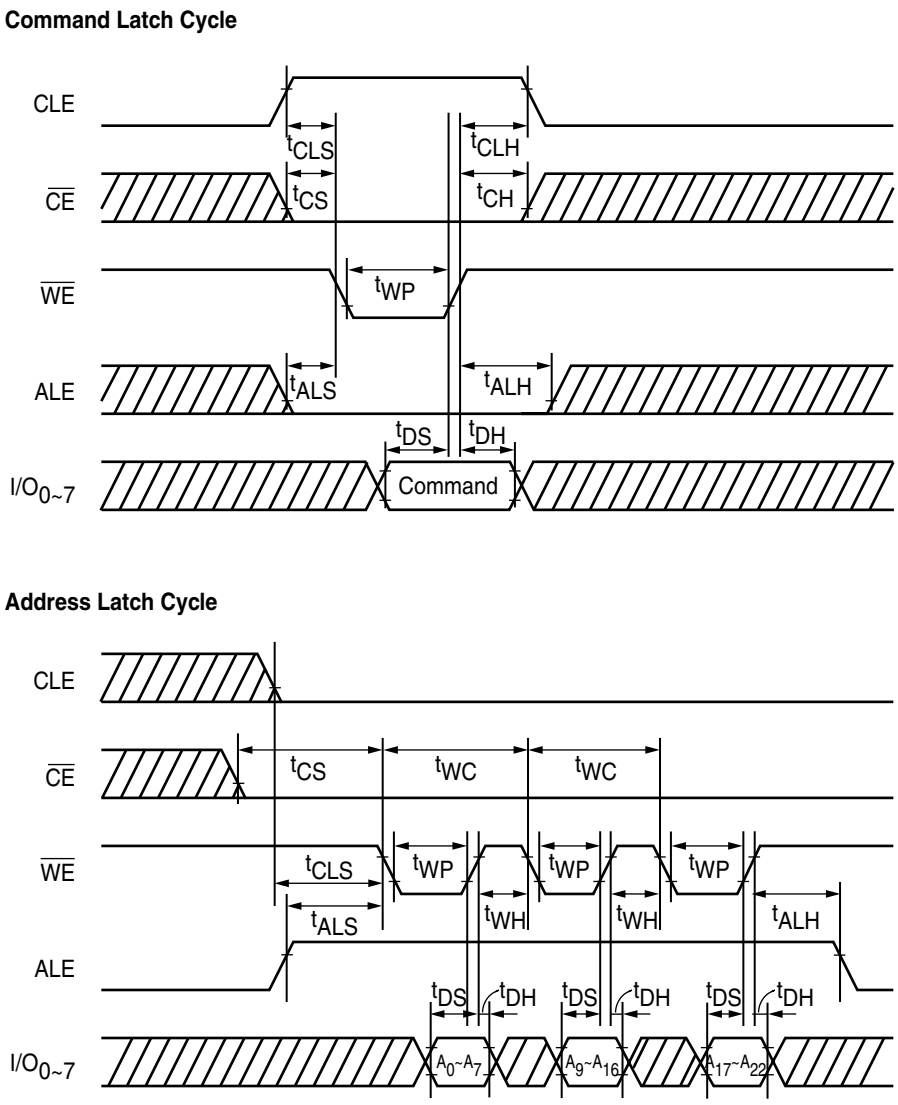
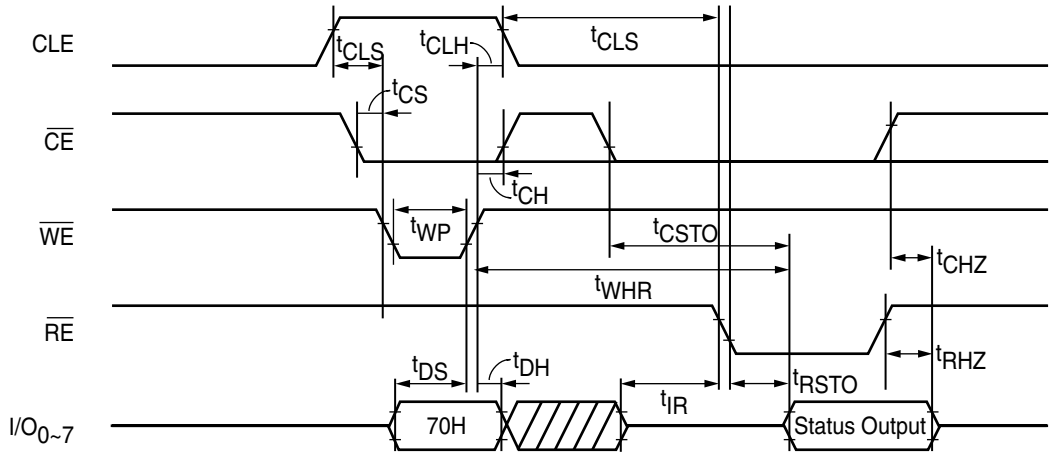


Figure 11 • *Command Cycle Waveforms*

For Read operations, the command is first written using CLE in the Command Latch Cycle, the starting address is written using ALE and WEn in the Address Latch Cycle, followed by pulsing REn for each successive read data as shown in Figure 11 on page 7. Both random reads and sequential row-read operations are possible with the Flash memory (Figure 12).

Status Read Cycle



Read1 Operation (Read one page)

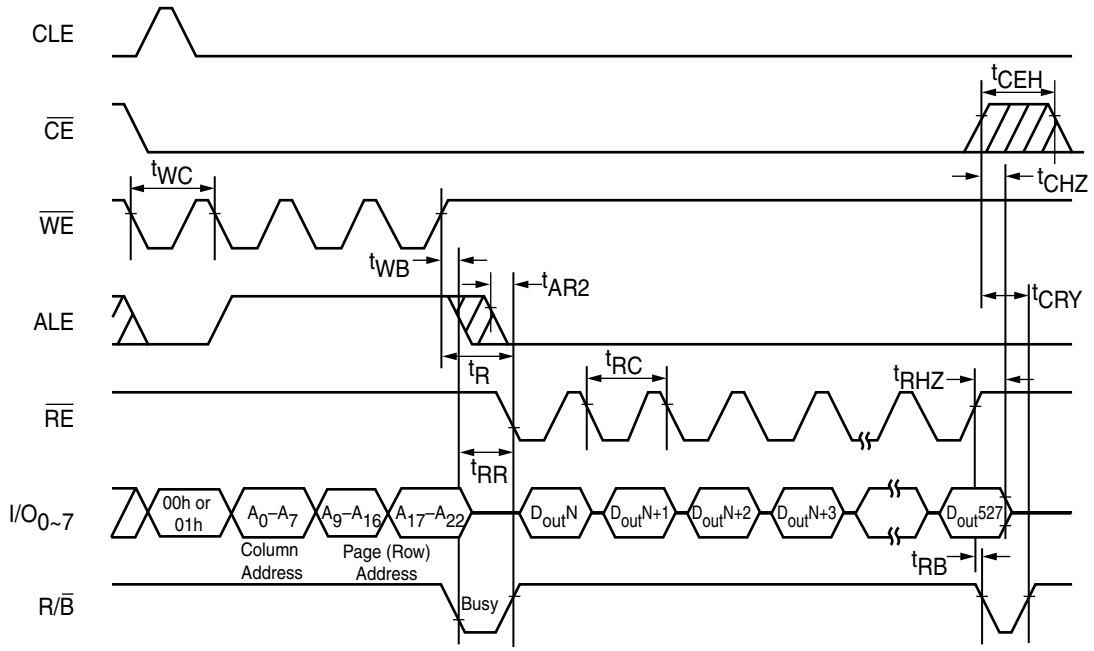


Figure 12 • Read Operations

For PAGE Program operations, the command is first written in the Command Latch Cycle, the starting address is written in the Address Latch Cycle, followed by pulsing WEn for successive writes of one up to 528 bytes of data, and terminated with another Command cycle, followed by a Read Status Command cycle to see if the entire program operation was successful (Figure 13).

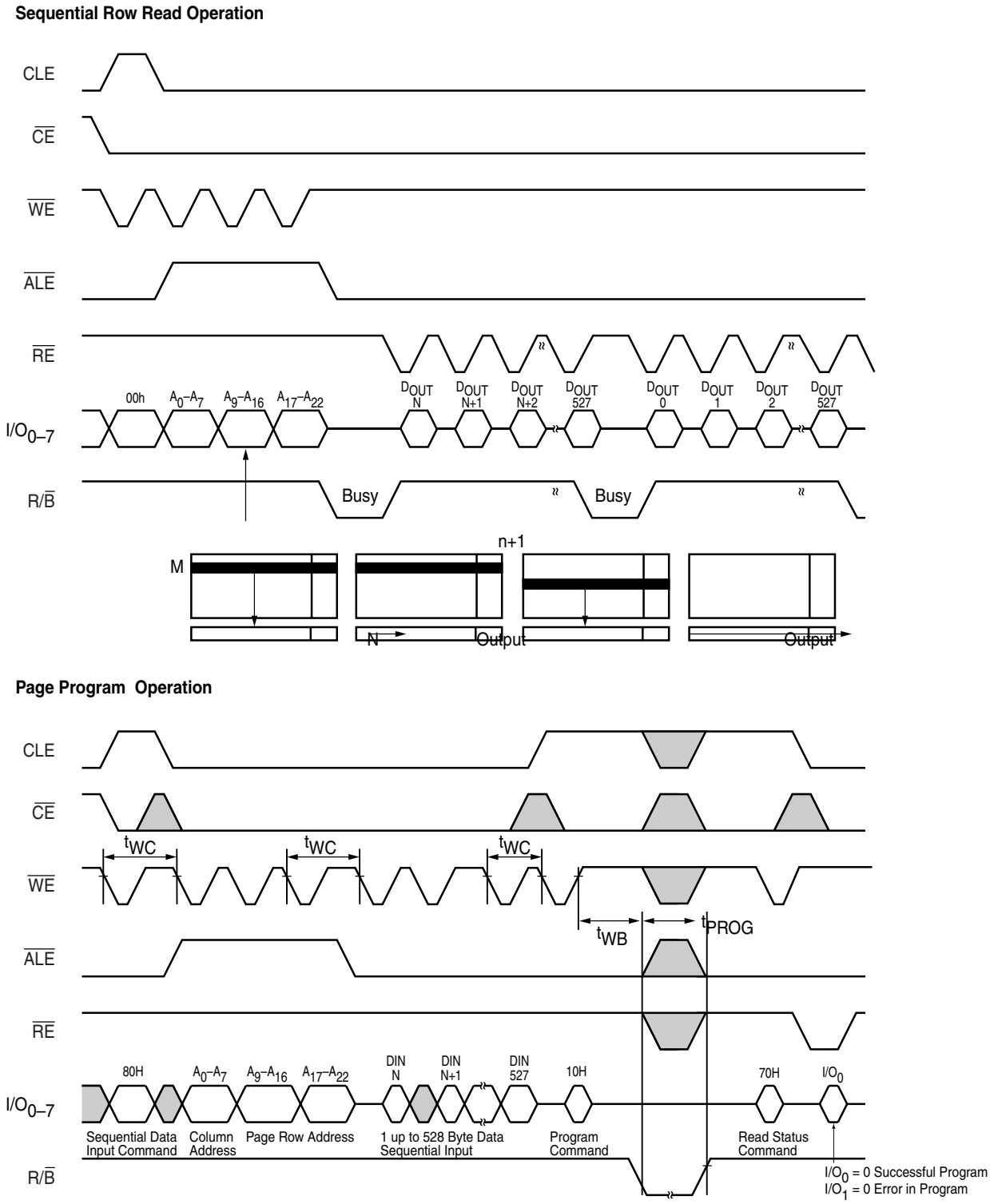


Figure 13 • *Page Program Operations*

A Block Erase operation is very similar to the PAGE Program operation.

There is also a Manufacturer and Device ID Read operation where a command latch cycle is followed by two successive reads by pulsing REn LOW.

All of the above operations are implemented in the A40MX04 interface block using a state machine driven by the FPGA clock at 14.725 MHz. The fastest serial page read access from the Flash memory is 50 ns minimum, which corresponds to a frequency of 20 MHz. However, the state machine that drives the Flash memory interface can use the 14.725 MHz system clock or a divided version of the clock to generate the signals to access the Flash memory. In the Rio PMP300, the actual access time used for the Flash was around 500 ns or approximately 2 MHz. Figure 14 shows the state machine for a serial one-page read (READ 1 in Figure 13 on page 9) operation from the Flash memory.

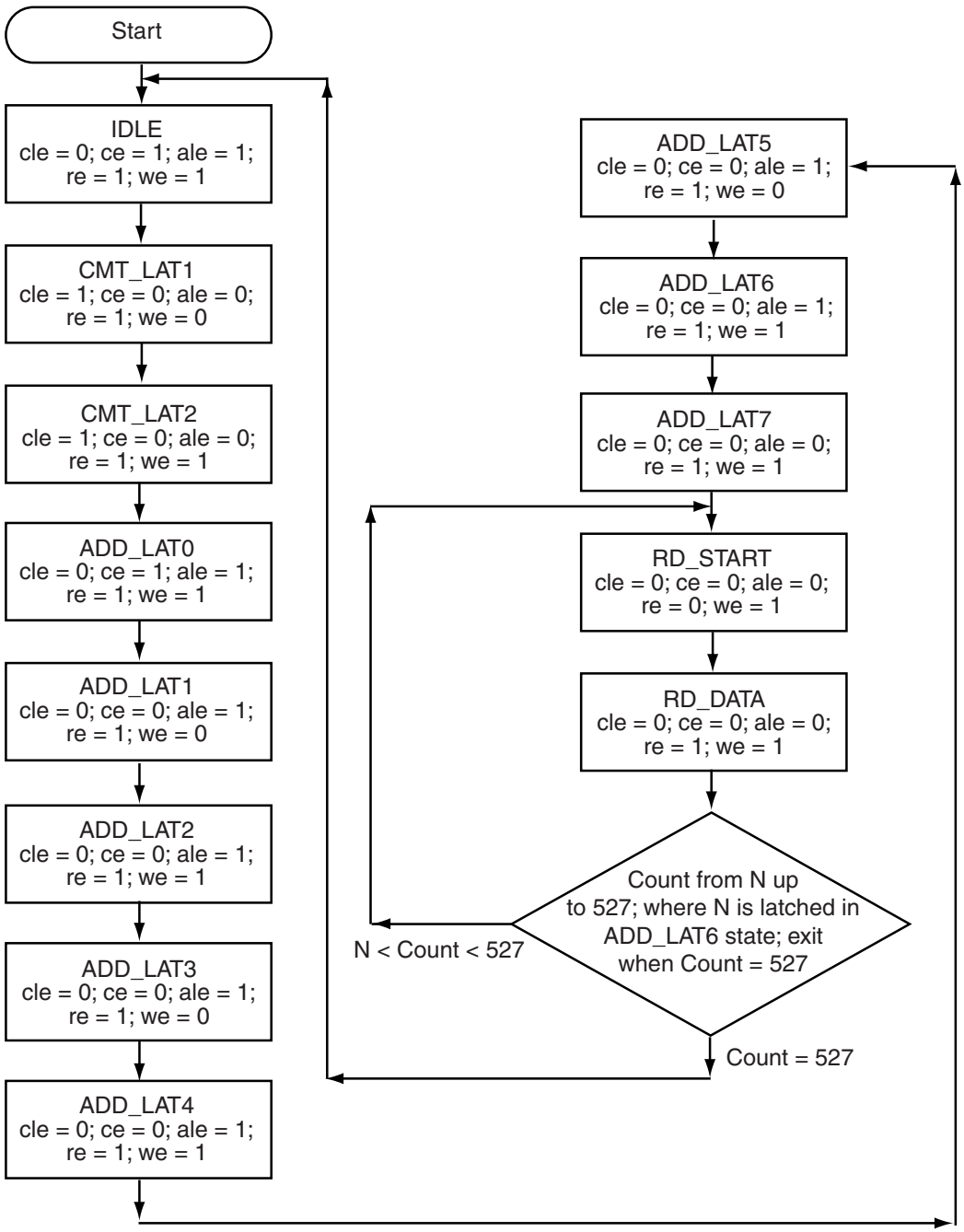


Figure 14 • *Serial One-Page Read State Machine*

The above state machine assumes that a clock frequency of 2 MHz is used and that R/B signal is HIGH for the entire duration of the serial page read. The operation starts with the state machine in the IDLE state. The state machine then transitions to states CMD_LAT1 and CMD_LAT2 where the command for the READ 1 operation is latched into the Flash memory. During these Command Latch states, the CPU writes a register in the FPGA, the register contents being driven on to the address/data bus, which are latched into the Flash memory on the rising edge of WE, in state CMD_LAT2. Due to hold requirements, CLE and CE are HIGH and LOW respectively until the next state, ADD_LAT0. The Address Latch states follow and the Column address is latched in on the next rising edge of WE in state ADD_LAT2, followed by the two portions of the Page Row address on the rising edges of WE in states ADD_LAT4 and ADD_LAT6, respectively. RE is pulsed successively and the read data from the Flash memory is available on the databus for latching by the Flash interface logic in the FPGA during state RD_DATA. An 11-bit counter in the FPGA logic increases for every read and when the count of 527 is reached the state machine exits to state IDLE again. For the read portion of the state machine, 4 state bits can be used to handle the 13 states required by this part of the state machine. Similar state machine logic was designed for the other operations required for Flash memory such as Program, Erase, Status Read, etc.

Smart Memory Interface Block

The Smart Memory card is an external Flash memory that requires a separate interface. This is used to increase the overall system memory to accommodate more songs in the player. The signal interface to the Smart Memory Card is very similar to the Flash Memory interface described in the previous section. The data rates from the Smart Memory are the same as for the Flash memory. The Smart Memory interface logic used 58 logic modules in the A40MX04 device for the implementation in the Rio PMP300.

CPU I/F Block

The CPU interface logic provides a path to communicate with the system CPU, a NECD78P064GC microcontroller chip with an integrated LCD controller. The CPU runs at 2.5 MHz and controls the operation of the entire system. The CPU wakes up the system and sets up the entire system for any required operation. The CPU has access to the system memory and manages the entire Flash memory configuration, programming, and access, including invalid block management. The CPU has an interface to the Micronas MP3 processor. The MP3 data read from the Flash memory is written to the MP3 processor by the CPU. The MP3 processor decodes the MP3 data and passes it to a DAC for analog output to the headphones.

The CPU interface logic in the MX FPGA comprises input latches to latch in the address, data, and control from the CPU and provide address, data, and control back to the CPU. The signal interface is shown in Figure 15.

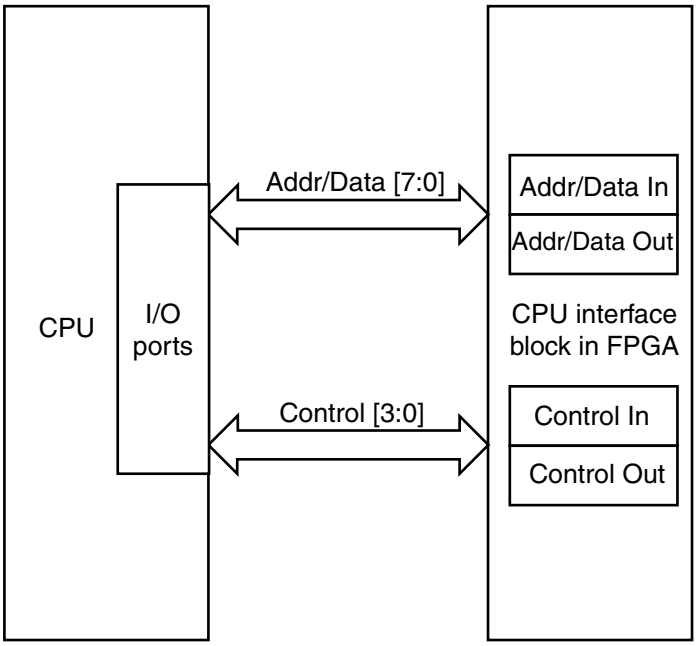


Figure 15 • *CPU Interface*

The CPU is an integrated microcontroller with ROM, RAM, LCD controller, and general-purpose I/Os (GPIOs). It communicates with the FPGA's CPU interface block through these GPIOs. The interface is a simple asynchronous interface comprising of address/data and a control port as shown above. The CPU writes address information intended for the Flash memory through the 8-bit address/data port, along with control information (that describes the kind of Flash operation to perform) on the control port. Some of the control port signals indicate what the information on the address/data bus is, such as a Flash address or data intended for the Flash memory. The logic on the FPGA side consists of a set of input latches for accepting the address/data and control information being written by the CPU. In addition, there are output registers in the FPGA interface logic that contain the data for the CPU when it reads the information (such as Flash memory data) back from the FPGA interface logic. The entire logic can be implemented with asynchronous combinatorial logic. The implementation of this CPU interface block on the Rio PMP300 used about 30 logic modules in the A40MX04 device.

MUX Block

The MUX block provides a data path for the data from either the parallel port interface or the CPU interface to either the Flash Memory interface or the Smart Memory interface as shown in the Rio PMP300 block diagram in Figure 7 on page 4.

When the player is in download mode, MP3 data from the parallel port interface is routed to the memories through the MUX block. In playback mode, data from either the Flash memory or Smart memory is routed to the CPU through the MUX block. The select signal for the MUX comes from bits that are set in the FPGA control register by the CPU. The MUX block is implemented using combinatorial functions in the logic modules of the FPGA.

Rio PMP500

The Rio PMP500 system is shown in Figure 16.

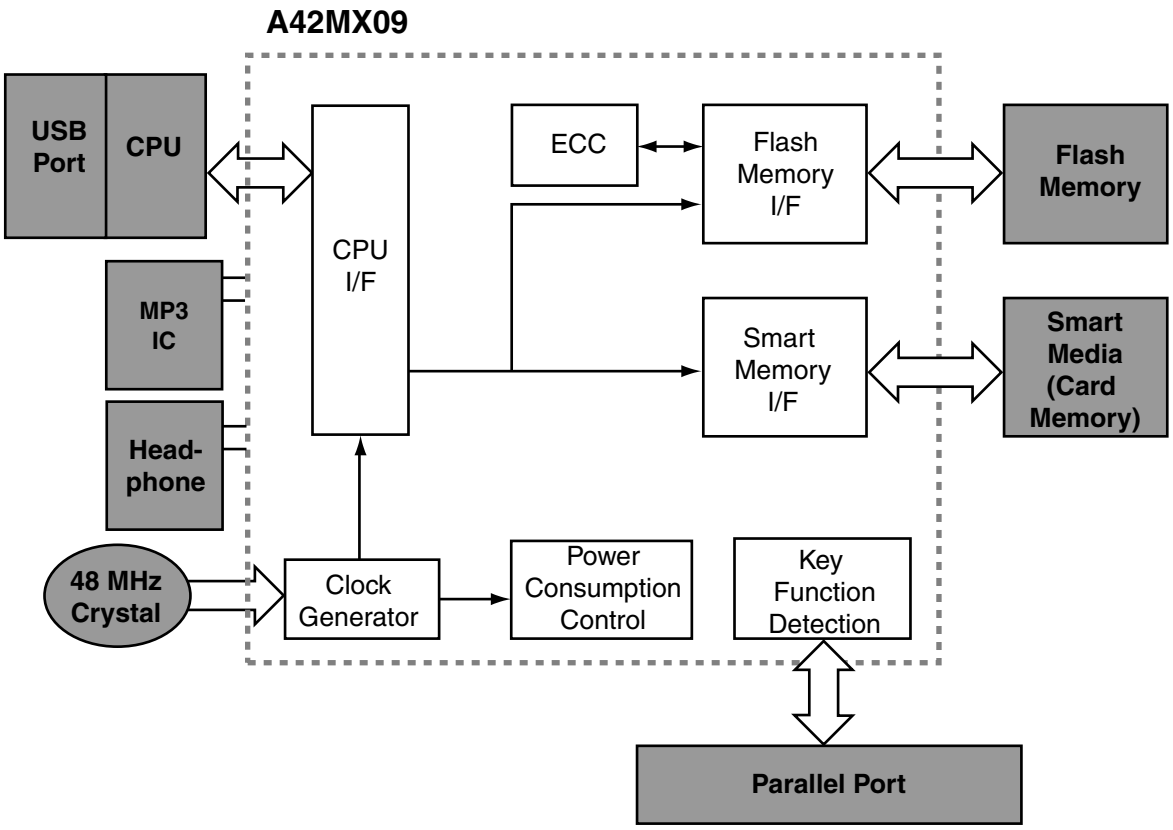


Figure 16 • *Block Diagram for A42MX09 in the PMP5000*

The Actel A42MX09 FPGA acts as the interface FPGA and houses the CPU interface, flash and smart memory interfaces with Error Correcting Code (ECC) support, clock generator logic, power consumption control logic and key function detection logic.

The CPU and Memory interface blocks are very similar to those on the Rio PMP300 described previously. In addition, ECC is implemented using a Hamming code function.

Hamming codes are an implementation of Forward Error Correction (FEC), which is the ability to correct an error when using data that may travel through a noisy media. The original data has extra check bits attached to each block to create a code word. These extra bits are calculated using a "Block Parity" mechanism that can be inexpensively implemented.

The Hamming code word is generated by multiplying the data bits by a generator matrix G using modulo-2 arithmetic. This requires a multiplier function that is easily implemented in the A42MX09 using the combinatorial functions of the logic modules.

The clock generator function generates various clocks from a 48 MHz input clock to drive the different portions of logic inside the A42MX09 device, as well as the rest of the system. The clock generation logic comprises dividers made up of flip-flops and combinatorial logic.

The keypad detection block is for changing the settings of the player. The keypad interface block typically comprises logic to detect the connection point (based on a key press) in an array and then translates this key press to the function associated with that key. Based upon this detection, the appropriate function is executed. The array is implemented with registers, latches, and combinatorial logic.

Next-generation MP3 Systems

Next-generation MP3 systems will improve over the existing systems by providing faster download, more music files for playback, and more options to the user from the interface/key panel with improved displays on the player. Faster download is accomplished through the use of a higher-speed parallel port mode (refer to IEEE 1284) or USB port. More music file playback can be offered through the use of higher-density Flash together with a local SDRAM-based song cache, allowing for additional options such as fast play, fast forward, shuffle play, programmed play, and so on. Other options could include an infrared beaming transceiver that would allow users to beam data back and forth from player to player or perform some remote-controlled operations. The entire system may be implemented in an Actel FPGA without the need for a CPU. An audio DAC controller logic block in the FPGA can provide an I2C interface to control the Audio DAC that takes in serial audio through the I2C interface and converts to analog audio to the headphones.

Other new compression schemes are emerging. Of those, the ATRAC3 standard seems to be gaining the most momentum. This standard has been adopted in the MS Walkman, VAIO Music Clip and the Network Walkman players by Sony Corporation. ATRAC3 is Adaptive Transform Acoustic Coding, which can compress CD music data by a ratio of 10:1 while retaining the sound quality of a CD. ATRAC3 is described below:

1. Encodes spectral coefficient data obtained through a signal analysis consisting of a band-splitting filter and Modified Discrete Cosine Transforms (MDCT) in an efficient way based on the characteristics of the input music signal.
2. Realizes a further bit rate reduction by adopting a joint stereo coding scheme, which takes advantages of the correlation between the two channels of stereo signals.

Using the above two techniques, a highly efficient compression of the music signal is accomplished.

Summary

This application note describes how the Actel A40MX04 and A42MX09 FPGAs were used to implement the main interface FPGA function in the Rio MP3 Digital Players, effectively meeting the performance and power requirements of the system. The highly flexible and efficient fine-grained architecture, the low-power characteristics, and performance of the MX devices were instrumental for their use in the Rio MP3 Digital Player systems. As a single-chip solution, MX devices substantially save on board space. The extremely small package size adds to the space savings. This, combined with its low-power characteristics, made the MX devices the best choice for use in the Rio MP3 Digital Players.

http://www.actel.com

Actel Europe Ltd.
Daneshill House, Lutyens Close
Basingstoke, Hampshire RG24 8AG
United Kingdom
Tel: +44-(0)125-630-5600
Fax: +44-(0)125-635-5420

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
USA
Tel: (408) 739-1010
Fax: (408) 739-1540

Actel Asia-Pacific
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Tel: +81-(0)3-3445-7671
Fax: +81-(0)3-3445-7668

5192667-0/9.00