Using Actel's A54SX08A FPGA to Interface a PowerQUICC Microprocessor to a MUSIC-IC LANCAM

Introduction

Content Addressable Memory (CAM) is a special memory designed to accelerate any application that requires extremely fast searches of list-based data such as database machines, image/voice recognition, or computer and communication networks. Typically, the CAM must interface with a microprocessor. The Actel SX-A family has all the features and flexibility required to implement this interface. This application note describes an interface design between a CAM (MU9C4480A/L LANCAM) from Music-IC Semiconductors (http://www.music-ic.com) and a Motorola PowerQUICC (MPC855T) microprocessor (http://www.motorola.com).

The Music LANCAM has an asynchronous 16-bit interface and supports three cycle types – each of which has a different cycle length. The PowerQUICC microprocessor has a 32-bit, synchronous interface. The different data widths and cycle lengths make it difficult for the MPC855T to interface directly with the LANCAM. Implementing a CAM controller in an Actel FPGA makes it easy to transfer data between the microprocessor and the LANCAM. The design can be implemented using an Actel A54SX08A FPGA, which automatically converts the different data widths and automatically handles the different cycle types.

The Actel A54SX08A FPGA has 12,000 system gates and supports system performance up to 250 MHz. SX-A devices greatly reduce design time by allowing for easy customization, enabling dramatic reductions in both design costs and power consumption, and further accelerating time-to-market for performance-intensive applications. For more information on the A54SX08A, see the SX-A data sheet located on Actel's web site.

There are many reasons the A54SX08A is suitable for this application, some of which includes: fast prototyping with easy design flow, ample timing performance required of the application, single chip for minimum board space, nonvolatility, and good utilization of the device.

What is a CAM?

In a conventional memory, data is stored in memory in specific locations called addresses. A CAM, however, is addressed by its contents, hence the name content-addressable memory, or CAM.* Data is supplied to the CAM via a special comparand register and the memory returns an address if a corresponding match is found. This enables extremely quick searches. The entire CAM is searched in a single clock cycle.

The comparison operation is done completely in hardware at the gate level and all locations are compared simultaneously with the comparand. A flag is set for each location that matches the comparand, and the locations of the flags are then read out. Because the operation is carried out in parallel for all locations on the chip, the results are available in a few tens of nanoseconds, regardless of the length of the list. Access is equally fast no matter what the order of the data in the CAM chip.

The search speed is thousands of times faster than sequentially searching a set of database records and even faster than microcode-based searching in a device controller. CAMs are extremely useful in networking applications because of the extensive use of various types of addresses. Since addresses can be very readily accessed in a CAM, the complex job of manipulating addresses to handle different formats is considerably simplified. In addition to its speed, a CAM provides a very compact implementation that minimizes board real estate for the search operation when compared with conventional static RAM searched by a microprocessor. Design and layout are correspondingly easier to do, and the associated software or firmware is much simpler.

CAM Controller Design

MU9C4480A/L LANCAM Description

The MU9C4480A and MU9C4480L LANCAMs are 4096 x 64-bit content-addressable memories with 16-bit wide I/O interfaces.* The MUSIC LANCAMs are ideal for network address filtering and translation applications in LAN switches and routers. The LANCAMs are also well suited for encryption, database accelerators, virtual memory, data compression, caching, image processing, and table-lookup applications.

The memory consists of static CAM organized in 64-bit data fields. Each data field can be partitioned into CAM and RAM subfields on 16-bit boundaries. Also, by using either of the two available mask registers, the CAM/RAM partitioning can be set at any arbitrary size between zero and 64 bits.

Active low signals are shown with a trailing lower-case 'n.'

The CAM subfield contains the associative data (used in compares), while the RAM subfield contains the associated data, which is not compared. For example, in LAN bridges the RAM subfield could hold port-address and aging data related to the destination or source address information held in the CAM subfield of a given location. In a translation application, the CAM field could hold the dictionary entries while the RAM field holds the translations. Each entry has two validity bits, the skip bit and the empty bit, associated with it to define its particular type – empty, valid, skip, or RAM. Vertical cascading of additional LANCAMs in a daisy-chain fashion extends the CAM memory depth for large databases. Cascading requires no external logic.

The contents of the memory can be accessed either randomly or associatively with a comparison. To use the LANCAM, the user loads the data into the comparand register, which is automatically compared to all valid CAM locations. The device then indicates whether one or more of the valid CAM locations contain data that matches the target data. The status of each CAM location is determined by two validity bits at each memory location. Loading data to the control, comparand, and mask registers automatically triggers a comparison. Comparisons may also be initiated by a command to the device. During automatic comparison cycles, data in the comparand register is automatically compared with the "Valid" entries in the memory array. Associated RAM data is available immediately after a successful compare operation. The status register reports the results of addresses. Two mask registers are available and can be used in two different ways – masking comparisons or data writes. The random access validity type allows additional masks to be stored in the CAM to be retrieved rapidly.

The device is controlled by a four-wire control interface and commands are loaded into the instruction decoder. A powerful instruction set increases the control flexibility and minimizes software overhead. Dedicated pins for match and multiple-match flags enhance performance of the device when the device is controlled by a state machine.

Description

The CAM Controller has two main blocks – the decoder block and the CAM_IF block. The decoder block contains the processor interface and the CAM_IF block has a state machine that generates the various LANCAM signals. In addition to these blocks, the CAM controller has read and write multiplexers as well as pipeline registers. Figure 1 on page 3 shows a block diagram of the CAM controller.

The decoder block decodes the address signal and generates various signals for the CAM_IF, which indicate the LANCAM cycle type and length (short, medium, or long). It gives information to the CAM_IF block whenever a TCO CT or TCO SC instruction is executed. This information is used by the CAM_IF block to snoop the data on the data bus. The TCO CT or TCO SC instruction changes the values of the segment controller source and destination counters and the terminal count of these counters inside the LANCAM. These counters determine the number of data read and write accesses to the LANCAM. The decoder block provides a registered version of the RWn signal to indicate either a read or write cycle. The block also generates a registered version of the address as an input to the write multiplexer. Finally, the decoder block generates the necessary control signals for the read multiplexer whenever a quick status read instruction is performed (indicated by setting bit 13 of ADDRESS to 1). The read multiplexer then selects the status signals (REG_MAn, REG_MFn and REG_FFn). The decoder block generates the TAn signal and associated tristate control signal.

The CAM_IF block uses the cycle-type and cycle-length information provided by the decoder block and generates the CAM control signals (En, Wn, CMn, etc.). The CAM_IF also generates the control signal for the write multiplexer and the bidirectional control signal for the DQ bus.

The CAM_IF block and the decoder block communicate with each other via three handshake signals. The decoder block pulses the signal CAM_START to indicate the start of a new cycle. The CAM_IF block indicates that it is ready for a new access by asserting the signal CAM_IDLE. The CAM_IF block asserts the CAM_IDLE signal when it is in the IDLE state and there is no access to LANCAM. Finally, the CAM_IF block pulses the signal CAM_TA to indicate the successful completion of the current cycle. The decoder block does not start a new access until the CAM_IF block asserts CAM_IDLE or pulses CAM_TA. When the CAM_IF block asserts CAM_IDLE or CAM_TA, the decoder block will assert the TAn signal to the processor to indicate the successful completion of the current cycle.

The CAM_IF block generates the LANCAM interface signals (En, CMn etc.) and the signals for controlling the write and read multiplexers. The CAM_IF interface block has a single state machine that generates all the LANCAM control signals, the internal control signals, and the decoder block handshake signals. The CAM_IF state machine has six different states as shown in Figure 2 on page 4.

These states are IDLE, START, CW_SM (command write state), CR_SM (command read state), DR_SM (data read state), and DW_SM (data write state). The state machine is in the IDLE state during reset. On receiving a CAM_START from the decoder block, the state machine goes into the

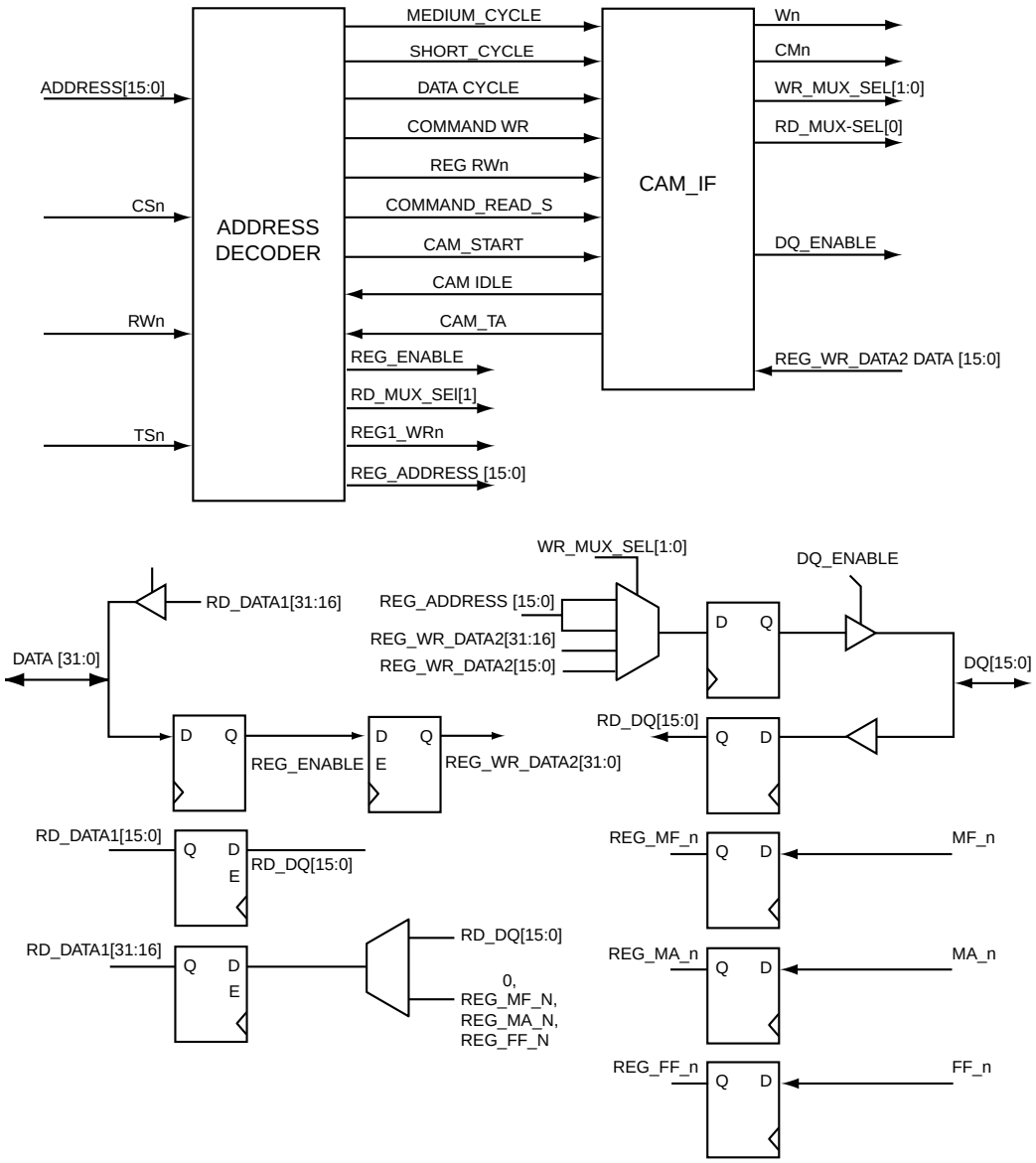

Figure 1 • *Block Diagram of CAM Controller (CAM_IFBlock)*

CYC_START state. Based on the value of DATA_CYCLE and REG_RWn (shown as the signal sel) the state machine transitions to the COMMAND_WRITE, COMMAND_READ, DATA_READ, or DATA_WRITE state. Each of these states consists of multiple states based on the length of the state machine and the number of words to be written. Upon completion of the current cycle in each of the four states, the state machine goes back to IDLE or CYC_START based on the value of the signal CAM_START.

During data cycles the user must keep track of the number of 16-bit words written, since the last data write is a long cycle. The CAM_IF block stores information when the processor executes a TCO CT or TCO SC write instruction. This information is used to determine the last data write based on the Destination Count Start and Destination Count End bits in the SC register of the LANCAM. Additionally, it determines the CAM/RAM bits in the control register of the LANCAM and also controls the write multiplexer and the read multiplexer.

Address Decoder

The address decoder determines if the address corresponds to a command cycle or a data cycle. Furthermore, the address decoder provides information to the CAM_IF block about the kind of LANCAM cycle – short, medium, or long. In addition, it gives information to the CAM_IF block about whether the opcode corresponds to a multicycle command. The address decoder stores information whenever an SPS or SPD instruction selects the memory array as the source or

destination. This information is then provided to the CAM_IF block during data cycles. The address decoder also gives information to the CAM_IF block when a TCO CT or TCO SC write command is executed. Additionally, the address decoder block also generates the TAn signal.

Implementation

The design presents a simple synchronous interface to the PowerQUICC MPC855T that can easily be customized and modified. Figure 3 shows the system-level block diagram. Table 1 on page 5 describes the processor interface signals and Table 2 on page 5 describes the LANCAM interface signals.

The address bus typically holds the LANCAM opcode. The LANCAM has two kinds of cycles (indicated by the CMn signal) – command (CMn low) and data cycle (CMn high). The upper nibble (bits 15 down to 12) of the LANCAM opcode is zero. The design uses this to indicate the nature of the cycle (data cycle or command cycle) as well as provide additional information. Bits 11 down to '0' of the ADDRESS bus are generally transmitted to the LANCAM without modification, while bits 15, 14, and 13 of the ADDRESS bus have special meaning and are not transmitted to the LANCAM. Bit 12 of the ADDRESS bus is not used. The definition of each of these bits is programmable and is defined in the .vhd file constant.

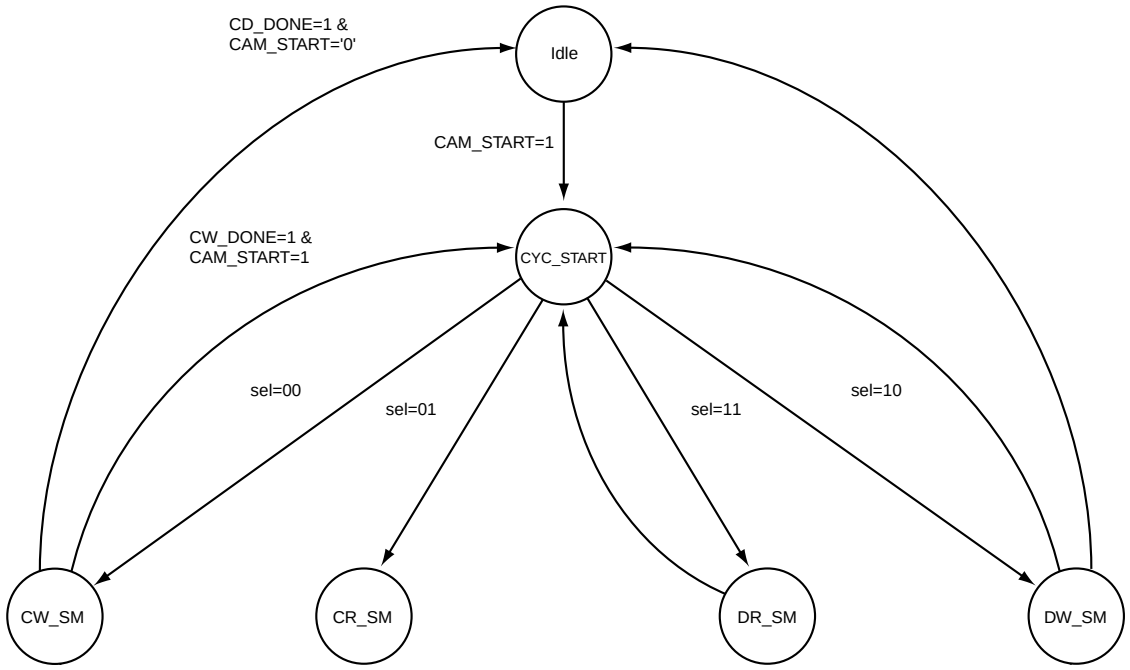

Figure 2 • *Simplified State Diagram of the CAM_IF Block*

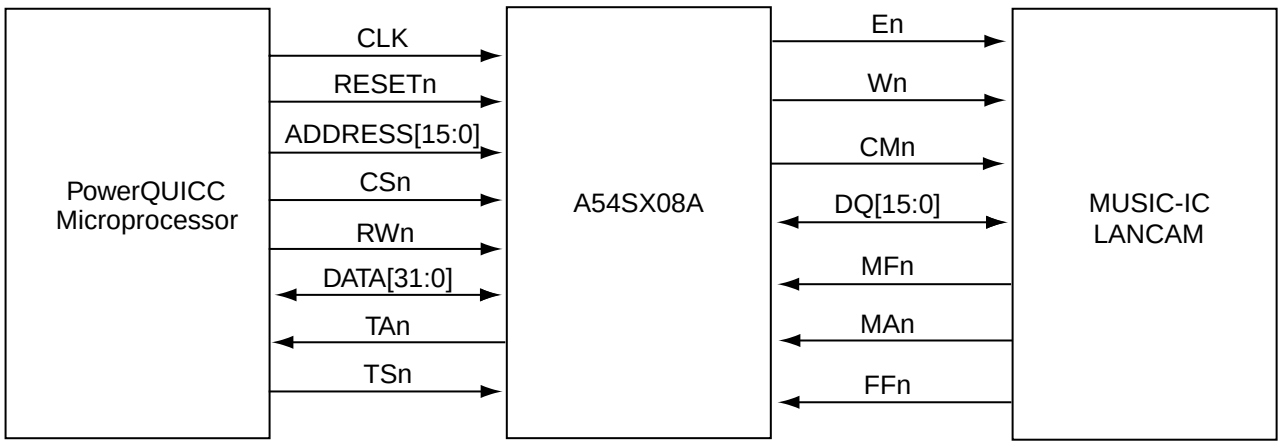

Figure 3 • *System-Level Block Diagram of the CAM Controller*

Bit 15 of the ADDRESS bus, when set to '1,' indicates a data cycle. When set to '0,' it indicates a command cycle.

Bit 14 of the ADDRESS bus, when set to '1,' indicates the opcode on the address bus has an associated operand. Some of the LANCAM commands like SPS M@aaaH take multiple LANCAM cycles. The first cycle has the command SPS (select persistent source for data reads) and the second LANCAM cycle has the address as defined by aaaH. Such a command is indicated by setting bit 14 of the ADDRESS bus to '1.' The value on DATA[15:0] is transmitted on the second cycle. In effect, a multicycle LANCAM command is converted into a single processor cycle.

Bit 13 is used to indicate a quick status read. The design registers the status signals (MFn, MAn and FFn) at the end of each access to the LANCAM. When this bit is set to 1 and RWn is '1' (read), then the FPGA puts the values of the signals on the data bus. The values of MFn, MAn, and FFn are output on bits 0, 1, and 2, respectively. The remaining bits of the data bus are set to zero. The values on the rest of the address bus bits are ignored and there is no LANCAM access. This feature enables a quick status read from the FPGA.

The processor data bus is 32 bits wide. The FPGA pipelines the data internally and transmits 16-bit data to the LANCAM.

Table 1 • MPC855T Interface Signals

Signal	I/O	Description
CLK	Input	50 MHz Clock
RESETn[1]	Input	Active-low reset signal
ADDRESS[15:0]	Input	16-bit opcode for 4480A/L. The opcodes are generally transmitted to the 4480A/L without any modification. If the opcode is part of a two-cycle command like "0804"H SPS M@aaaH. The lower 16-bits on the data bus are transmitted to 4480A/L on the next cycle. If any of the bits in the upper nibble of the address is a '1,' the opcode is not transmitted to 4480A/L. This feature can be used for data writes and reads.
CSn	Input	Active low Chip Select
RWn	Input	READ/WRITE signal. RWn low selects a write operation and RWn high selects a read operation.
DATA[31:0]	Input/Output	Bidirectional data signal. For multi-cycle commands, the lower 16-bits of the data bus are transmitted on the next LANCAM cycle.
TAn	Output	Transfer Acknowledge
TSn	Input	Transfer Start

Note: *1. Active low signals are shown with a trailing lower-case 'n.'*

Table 2 • MU9C4480A/L Interface Signals

Signal	I/O	Description
En	Output	Active low Chip Enable
Wn	Output	READ/WRITE signal. RWn low selects a write cycle and RWn high selects a read cycle.
CMn	Output	Data/Command Select
DQ[15:0]	Input/Output	DQ[15:0] are used to convey data, commands, and status to and from the LANCAM.
MFn	Input	Match Flag
MAn	Input	Multiple Match
FFn	Input	Full Flag

Operation

The start of a processor cycle is indicated by the assertion of TSn, CSn, and by having valid data on the address bus and RWn signal. In case of a data write or multicycle command write, the next cycle has valid data loaded on DATA[15:0]. The FPGA decodes the address and generates the necessary signals to interface with the LANCAM. At the end of the access, it asserts TAn.

Single-cycle Command Writes/Reads

For single-cycle commands, the processor loads the opcode on the address bus and asserts CSn and RWn. The FPGA decodes the address and starts the transfer to/from the CAM. The FPGA interfaces with the processor using the TAn and TSn signals.

Multicycle Command Writes/Reads

For multicycle commands, the processor loads the opcode onto the address bus and asserts CSn and RWn. For a multicycle command write, DATA[15:0] also has the data for the second cycle. For command reads, the FPGA interfaces with the processor using TAn and TSn signals.

Data Writes

The processor loads the opcode onto the address bus and asserts both CSn and RWn. DATA[31:0] has the data that needs to be written to the LANCAM. On the next clock cycle, the data bus has the upper two words of data that need to be written to the LANCAM.

Data Reads

The processor loads the opcode onto the address bus and asserts CSn and RWn. The FPGA reads the data from the LANCAM and then presents the data to the PowerQUICC.

Utilization and Performance

Table 3 shows the area utilization in an A54SX08A. The percentage utilization is also shown in parentheses. Table 4 shows the key performance numbers.

Table 3 • Utilization in A54SX08A

Area Used (Percentage)	Total Available Cells
229 Sequential Cells (89.45%)	256
145 Combinatorial Cells (28.32%)	512
374 Total Utilization (48.70%)	768

Table 4 • Performance in A54SX08A Std. Speed Grade

Description	Post Layout	Required
Clock Frequency	98 MHz	50 MHz
External Setup for Processor Interface Signals	3.1 ns	20 ns
Clock-to-Out for Processor Interface Signals	11 ns	20 ns

Conclusion

There are numerous reasons why the A54SX08A is suitable for this application. The combination of Actel's development tools and the A54SX08A device allows fast prototyping with easy design flow. The superior system performance of the A54SX08A provides ample timing margin for this application. Furthermore, the A54SX08A is a single chip solution that occupies minimum board space. The device is also nonvolatile, which means there is no need for a startup bitstream, eliminating the possibility of configuration data being intercepted and copied or reverse engineered. This also prevents in-system errors and accidental data erasures that can otherwise occur during download.

The Actel A54SX08A provides a flexible interface to communicate with the LANCAM. The implementation can be easily customized to provide an interface between various microprocessors and CAMs. The automatic handling of various LANCAM cycle lengths and the conversion of multicycle commands into single-cycle ones frees up the microprocessor and improves the overall system performance.

References:

Using the MU9C1965A/L LANCAM ® MP in LAN-ATM Edge Switches

http://www.music-ic.com/literature

MU9C4480A/L LANCAMs ® Data Sheet

http://www.music-ic.com/literature

PowerQUICC MPC Documentation

http://www.motorola.com

http://www.actel.com

Actel Europe Ltd.
Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401590

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

5192677-0/7.01