PLD Integrates Dedicated High-Speed Data Buffering, Complex State Machine, and Fast Decode Array

Tarif Arabi, Applications Manager
Dave Lewis, Strategic Marketing Engineer

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
(408) 721-5000

Introduction

Many designs contain both complex synchronous logic and fast asynchronous logic which must be integrated in Programmable Logic Device(s) (PLD's). For example, some systems have a need for a specialized memory controller, usually a big state machine, as well as a need to decode high speed combinatorial signals such as chip selects, RAS and CAS, bank switching signals, etc. If the PLD is to be synchronized with the rest of the system, additional flip-flops will also be needed to buffer input signals. Implementing such designs in standard PLD's can be difficult since a PLD must meet the conflicting demands of control logic complexity and high-speed data buffering/decoding. In situations like this, the designer may wish to try a unique solution like the MAPL™244 PLD from National Semiconductor.

This paper utilizes a multiport buffer memory controller application to illustrate how to take advantage of the MAPL244's integrated state machine/control logic array, fast decode array, and high speed input data buffers.

The Application

A functional block diagram of the buffer memory controller is shown in Figure 1. Three ports (PORT2, PORT1, & PORT0) share two banks of buffer SRAM. These ports share a data bus via latching transceivers, but address the memory through an address mux which is selectable by the buffer memory controller implemented within the MAPL244. All three ports are similar and have a lock mode that denies memory access to the other ports. In addition, PORT2 also has burst mode capability. Port requests for memory access are double-buffered, for reduced metastability, and ports are granted access by a small, but fairly complex state machine. A small amount of additional logic performs memory bank switching and outputs port access status.

A typical sequence of events starts with the state machine in the idle state when a port, say PORT0, requests memory access. In the next clock cycle, the system is informed that PORT0 has been granted memory access. Next, the appropriate write or read enable signals are given in the following clock cycles. If PORT0 has locked access, then memory access by other ports are denied until PORT0 cedes access. If, however, PORT0 does not lock access, then a memory access request by another port will be granted. PORT2 functions similarly but also has burst mode capability.

Application Problems

The first problem encountered was to integrate the state machine into a PLD. Although the state machine contains only 14 states, it has many transitions making it fairly complex. The PLD also had to accept future changes to the state machine such as adding another port or integrating new memory control features. These changes must fit into the same PLD without altering the placement of existing pins. Traditional PLD solutions such as utilizing multiple high-speed PAL®'s or a single high density PLD were considered. The multiple PAL solution was rejected based on this criteria. It consumes a lot of board area, especially when the memory access controller state machine is partitioned across the PAL's. In addition, significant changes to the state machine would require major

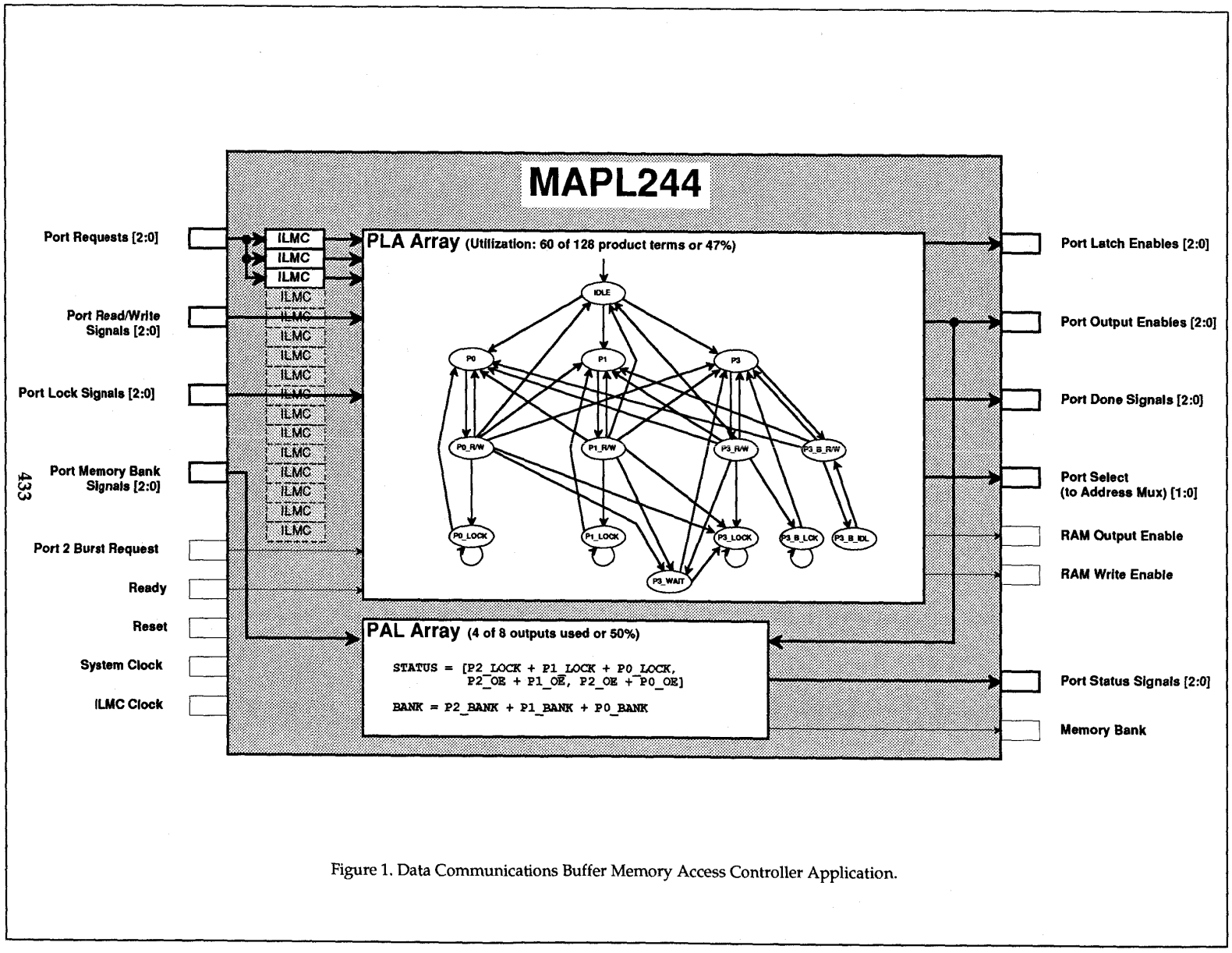

Figure 1. Data Communications Buffer Memory Access Controller Application.

board layout modifications. On the other hand, using a high density array-based complex PLD solves the integration problem and retains PAL simplicity, but would have not fit the design if an additional port were added. Also, some complex PLD's have trouble with consistent pin placement when design changes are made.

There is also a second problem: signal timing. The problem occurs at the port memory access request inputs to the state machine. These signals are asynchronous and require the PLD to have very short setup times. It was determined that input buffers would be needed, so some D flip-flops had to be added buffer the port request signals. Since it would be useful to integrate these too, a register-rich, cell-based FPGA seemed promising. The design is a good fit for FPGA's that can handle the many state transitions of this state machine. FPGA's can, however, be expensive and while future modifications to the design will likely fit, they often impact development time as well as signal timing. Since signal timing issues are critical in this design and cost is always a big issue, another PLD alternative was desirable.

MAPL244 Solution

The MAPL244 (figure 2) is a medium density, 32 I/O EECMOS PLD that allows easy integration of complex subsystems consisting of synchronous and asynchronous elements. It integrates a large paged-PLA (programmable AND *and* programmable OR arrays) structure for synchronous control logic, a PAL (programmable AND array only) block with combinatorial I/O to handle asynchronous and combinatorial decode functions, and 16 Input Logic Macrocells (ILMC's). Each of the 16 Input Logic Macrocells (ILMC's) (see Figure 3) can individually be configured as a latch, register, or double register.

In this application, three of the 16 ILMC's are configured as double buffers and are used to capture the port memory access request signals. The first register is clocked by the falling edge of the clock while the second register is clocked by the rising edge. This double buffer setup alleviates metastability problems since the external data is synchronized to the state machine in two successive steps. The single latch and register modes could have been used and in fact, different modes can be configured for each input.

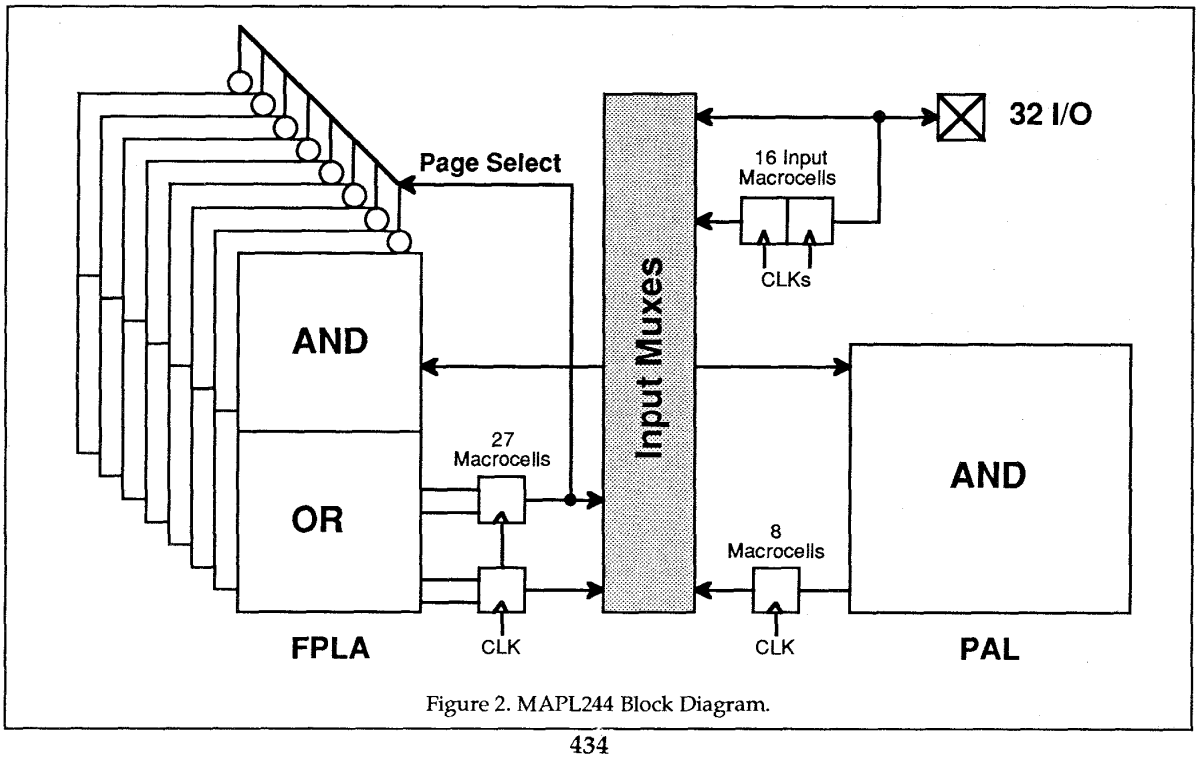

Figure 2. MAPL244 Block Diagram.

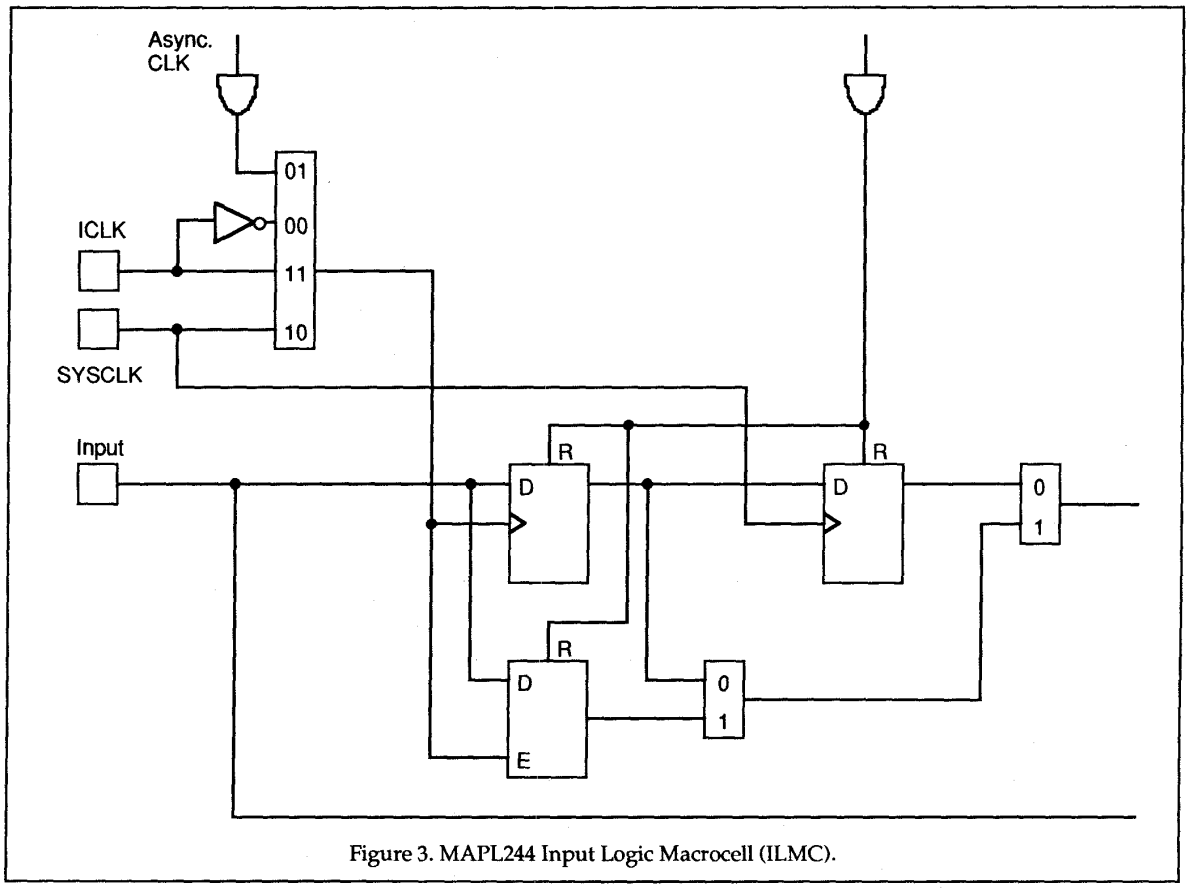

Figure 3. MAPL244 Input Logic Macrocell (ILMC).

The state machine, which is small but contains many transitions, is implemented in the PLA and consumes less than half of the product terms. Adding another port (PORT3) would bring PLA utlization to only 60%. D flip-flops are used at all outputs, though DE, JK, RS, and T flip-flops could be used with equal performance since both JK and DE flip-flops are implemented in hardware (see Figure 4). The PLA is split into pages of which one is active at any given time so that power consumption is only 180mA worst case at 25MHz for the entire chip. The paging feature is performed automatically by the fitter software and is transparent to the user during the design and the operation of the system. Since the I/O have access to all pages, changes in design do not cause fitting problems associated with other multiple array architectures.

The PAL array is used to generate the memory bank signal and the access status signal bits. This PAL array is essentially a 10ns GAL24V8 and can implement functions that would normally be implemented in a 10ns GAL22V10 or GAL20V8. These functions may be either independent or dependent of the state machine and might include address decoding, RAS/CAS generation, statebit decoding, error detection and correction, etc. Note that I/O pins associated with the PAL array that are not used or whose macrocells are used for buried feedback are free be used as inputs to the PAL or PLA arrays. This feature helps conserve I/O.

Conclusion

Many applications, such as memory controllers, require both complex synchronous and fast asychronous logic elements. Since traditional PLD solutions may not be well-suited for these applications, the designer may wish to utilize unique alternatives such as the MAPL244.

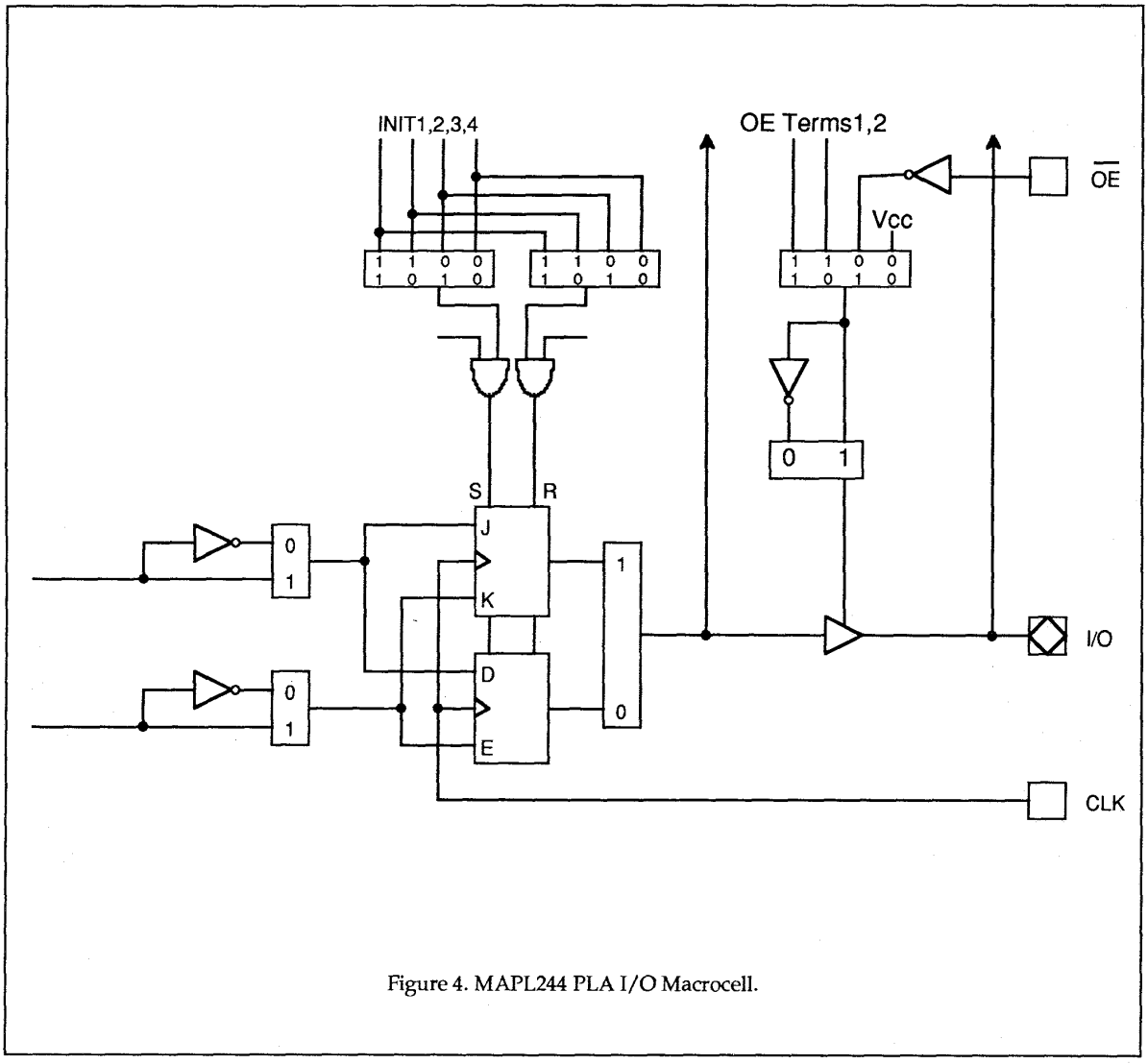

Figure 4. MAPL244 PLA I/O Macrocell.