Abstract:

A 30ns Mappable Peripheral Memory Subsystem (MAP) will be described. The MAP significantly enhances system performance by integrating on the same chip, 128K bit EPROM for program store, 32K bit SRAM for data store and a Programmable Mapping Decoder (PMD) for address decode and mapping. The PMD is integrated into the memory decode logic without adding to the access time. The decoder facilitates address mapping within a 2MB address space. The MAP is ideally suited as an extension of peripheral memory for high speed Digital Signal Processors (DSP's), microprocessors and microcontrollers. The 290 mil sq. chip is implemented in a 1.2um CMOS EPROM process.

I. INTRODUCTION

In a typical microprocessor based system the address map is partitioned into blocks of program, data and other special segments i.e., I/O. An address decoder is used to facilitate this mapping by selecting the appropriate memories (EPROMs, SRAMs, etc.) corresponding to the memory map. In addition to this, the memory might have to be significantly enhanced system non-contiguous address space.

Regardless of the method chosen to implement the decoder, some compromises will be incorporated that affect system performance, board space and cost. Since the decoder is in the memory access path, the total memory access time is the sum of the decoder delay and the access time of the memory. For example, to achieve a 30ns total access time, an 8ns decoder can be used with a 20ns memory allowing 2ns for printed circuit board (PCB) interconnect delay. A high-speed memory requirement have become very prominent with the emergence of fast microprocessors and microcontrollers (i.e., 30MHz clock rates).

Integrating the decoder with the memory on the same chip results in a number of advantages. First, there is an improvement in speed: the decode delay is absorbed in the total access time. This is done by placing the PMD in the fast path of the memory, i.e., Y-select path. Thus, the PMD does not add to the memory access time. Second, the PCB interconnect delay between the decoder and the memory is eliminated. Finally, the board space is reduced.

II. ARCHITECTURE

The MAP is structured as a series of blocks to achieve a very flexible and highly configurable circuit for general purpose applications (Fig. 1(a)). The EPROM is subdivided into 8 blocks and the SRAM is subdivided into 2 blocks. These memory blocks can be considered as separate memories with dedicated chip selects. The on-board PMD selects the appropriate block based on the incoming address. The PMD actually functions as an address comparator for each block that was previously programmed into the PMD. If no block is selected by the address, both EPROM and SRAM are powered down and the outputs are disabled. This allows other devices to drive the data bus. In addition to selecting internal blocks of memory, the PMD can also be programmed to select other devices using Chip Select Outputs (CSO's).

III. PROGRAMMABLE MAPPING DECODER (PMD)

The PMD is implemented similar to a PAL device [1], see Fig. 1(b). All the block decode addresses are connected to the AND plane. There is only one output per AND gate (there is no OR plane). Each AND gate output either selects a block of internal memory or in the case of CSO's (CS00-7), selects a number of blocks of external memory. Addresses A11-A20 are block decode addresses, EPROM select outputs ES0-ES7 (ES outputs) select 1 of 8 available EPROM blocks. SRAM select outputs RS0-RS1 (RS outputs) select 1 of 2 available SRAM blocks. This architecture enables the MAP device to be configured and compatible with virtually any system address map up to 2M-bytes. Complicated address maps of microcontroller systems can be fully utilized by programming blocks of EPROM and SRAM to be addressed by the various portions of the system address map.

The fast decoder design was achieved using the following circuit techniques: PMD array is based on a dedicated source architecture, that is, each product term bit-line has its own dedicated source line (see Fig. 2). A current-limiting device (M3), in the S/L path, is used to set the low level of the S/L voltage. The low level is automatically clamped to one Vth by feeding the amplified B/L node to the gate of M3. Therefore, the low level is independent of the number of active pull-down devices, this effectively avoids the "Super-Zero" problem of programmable decoders and reduces the B/L swing [1]. The high level is clamped at Vcc-Vth by using a self-biased p-channel pullup M2. In addition to reducing the B/L swing (for high speed), this architecture has much lower power dissipation per product term.
limited to 100µA per product. The total decoder delay (address to YS-MUX output) is 4.8ns.

IV. COMMON I/O

Each output of EPROM and SRAM is connected to a common data line. As a result, the data line is fairly long and capacitive (Fig. 3). The fast data line drive scheme is achieved with the following circuit technique: INV1 and INV2 act as trip inverters for the current-mirror sense amp (CMSA) [2], and also equalize the data line at Vcc/2 from one end. The last trip inverter INV4 helps equalize the data line from the other end. TriiNV is a high-drive tri-stateable inverter which is tri-state during equalization. Therefore, it is not drawing any power. Once the CMSA output is set, INR clock releases INV1 and INV2, then, EOE (EPROM read) releases TRiNV. TRiNV is required to move the large data line capacitance (CDL) by only 100-200mV. After this, the final trip inverter INV4, which is released by the BUSR clock, drives a small capacitance CDAT. This scheme provides a very fast (~2ns) data line drive technique without dissipating a lot of power.

V. FEATURE SUMMARY

The 128K EPROM is based on the split gate EPROM technology for high density and very high speed [2]. The 32K SRAM is based on the industry standard full CMOS 6-transistor cell. Both the EPROM and SRAM can be configured for a byte wide or word wide mode. The high speed on-chip common I/O bus between the EPROM and SRAM is contention free. This is achieved by using address transition detection (ATD) and clocking to trip-state the drivers of the EPROM and SRAM while the PMD is decoding. An added advantage of this architecture is that the PMD code can be secured; thus, making it hard to directly copy the part. This is done by providing a security bit (one EPROM cell) which disables the PMD data from being accessed directly (MUX L in Fig. 1(a)). The MAP is implemented in a 1.2µ CMOS process. The typical access time is 30ns including decoding, and the power dissipation is 325mW.

REFERENCES


Fig. 3 Common I/O and data-line drive circuitry.

Fig. 2 PMD array architecture.