

An SRAM-Programmable Field-Configurable Memory

Tony Ngai^ψ, Jonathan Rose, and Steven J. E. Wilton

Department of Electrical and Computer Engineering, University of Toronto
10 King's College Road, Toronto, Ontario
CANADA M5S 1A4

^ψ
Now with AT&T Bell Laboratory, FPGA Development Group, Allentown, PA., U.S.A.

Abstract

This paper describes the design and implementation of an SRAM-Programmable Field-Configurable Memory (FCM), which has the flexibility to form over two hundred different memory configurations, each with up to four individual memories. The prototype Field-Configurable Memory has four 1Kb memory blocks, each of which can be configured into four different aspect ratios. It requires just 40 configuration bits and is only 38% larger and 46% slower than the ASIC memory upon which it is based. User memories implemented on this chip require from 16 to 23 times less area than if they were implemented on a Xilinx 4000 series memory architecture. Although this is a stand-alone FCM implementation, the design can be embedded as part of an FPGA.¹

1. Introduction

Field-Programmable Gate Arrays (FPGAs) are now widely used to implement digital logic circuits, but most have little or no on-chip memory. A single FPGA will soon be large enough to integrate an entire digital system, and as most systems are composed of both logic and memory, FPGAs with significant memory capacity will be needed. The key element of such memory will be its ability to accommodate various numbers and shapes of memories, because target applications have widely varying memory requirements. We call a memory with this capability a Field-Configurable Memory (FCM).

On-chip Field-Configurable Memory will also provide significantly higher memory bandwidth than off-chip memory because the pin count limitations prevent wide off-chip memories and I/O pad delay limits off-chip access time.

Several FPGAs vendors already offer limited memory capability [Brit93] [Hsie90] [Plus89] [Marp92] [Smit93]. Although each of these architectures individually have either good performance, memory density or flexibility, none can achieve all three features at the same time [Ngai94]. In this paper we present the design and implementation of an FCM

that can implement memories that are fast and dense yet is flexible enough to support a wide range of memory configurations. In the next section we describe our general notion of a Field-Configurable Memory. Section 3 describes the specific architecture of our prototype FCM and Section 4 gives results measured from the implemented chip.

2. Overall Architecture of an FCM

In the remainder of this paper, we will refer to the set of memories required by an application circuit as a *logical memory configuration*. Each independent memory within a logical memory configuration will be referred to as a *logical memory*.

The basic FCM structure consists of several Basic Memory Blocks (BMBs) connected by a programmable interconnect structure as illustrated in Figure 1. The BMBs are the physical memory units which can be used individually or combined together to form logical memories. The interconnect structure provides all the necessary connections between the BMBs and the external world (I/O pads for a stand-alone FCM or other programmable interconnects for a mixed FCM-FPGA design).

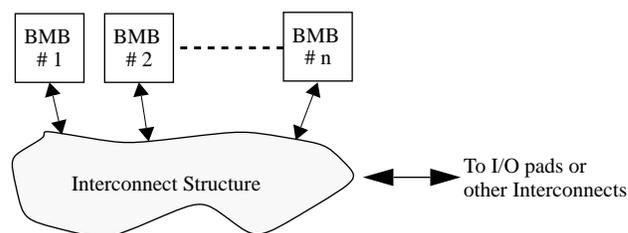


Figure 1. Overall Architecture of an FCM

There are two ways that BMBs can be combined: either “vertically” to form deeper logical memories, or “horizontally” to form wider ones. Figure 1 shows an example using two 256x4 BMBs. Figure 1a shows how the two BMBs can be used to form a 256 x 8 logical memory and Figure 1b shows how a deeper 512 x 4 logical memory can be formed. In order to increase the overall flexibility of

1. This work has been supported in part by a grant from MICRONET.

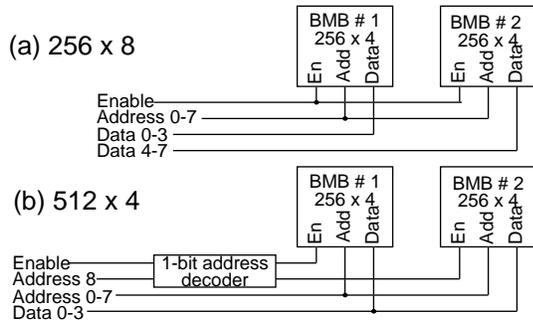


Figure 2. Two Different Ways to Group Two 256x4 BMBs

the FCM, we believe that it is important to make each BMB configurable into different aspect ratios [Wilt95].

3. Architecture of the Prototype Chip

In this section we describe the architecture of a prototype stand-alone FCM, called FiRM (for **F**ield-**R**econfigurable **M**emory). It consists of four 1K bit BMBs and thus can implement up to four logical memories. The memory is based on a synchronous SRAM design for ASIC memory from Bell-Northern Research provided through the Canadian Microelectronics Corporation. The aspect ratio of the base memory array is 128 x 8 bits giving seven address and eight data lines. The following sections describe the details of the BMBs, and the routing architecture that programmably joins the BMBs.

3.1 The Basic Memory Block

Each BMB can be programmed to take on an aspect ratio of 1K x 1, 512 x 2, 256 x 4, or 128 x 8. This flexibility is achieved using a mapping block surrounding the basic memory array as shown in Figure 3. The core of the mapping block is a pass-transistor network that connects the fixed-width 8-bit data bus (M bus) from the memory array to the variable-width data bus (D bus) emanating from the BMB. The two configuration bits (Cbit0 and Cbit1) are used to select one of the four output widths. The address lines A0 to A6 are connected to the memory block to select one of the 128 bytes. The mapping logic then decodes the upper address (A7 to A9) to connect the external D bus to the corresponding portion of the internal M bus.

Figure 4 illustrates the switch settings of the mapping block when the BMB is in the 1Kx1 and the 256x4 mode. The horizontal lines are the output D bus, and the arrows show how these signals are connected to the fixed M bus. The connections are made by turning on the appropriate

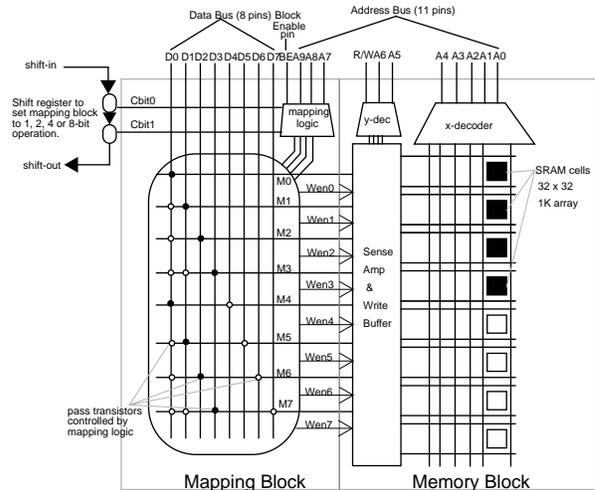


Figure 3. Block Diagram of the BMB

group of pass transistors indicated by the solid circles in the two diagrams.

To support a write operation into the array when the D bus width is less than 8 requires a way to write into only part of the M data-path without changing the remainder. For this reason the base SRAM array provides separate write enable lines for each bit of the base memory word. For example, as shown in Figure 3, the solid circles, arrows and squares represent the 256x4 configuration. When the address is between 0 to 127, the Wen0-3 signals are activated so that the write buffer updates only the top four SRAM cells but not the bottom four.

3.2 Routing Architecture

The purpose of the routing structure is to connect several BMBs to form one logical memory, making them either deeper or wider than the native capability of the BMB itself. The routing structure also provides connectivity to the I/O.

The FiRM chip employs a hierarchical routing structure, as illustrated in Figure 5. Two sets of two BMBs are grouped at the lowest level of the hierarchy. The I/O pads are connected to the top of the hierarchy, providing a constant two-pass-transistor path to all BMBs. This means that the access time of different logical memory configurations should be consistent and predictable.

The routing architecture for an FCM can be far more area-efficient than that of a general-purpose FPGA switching a

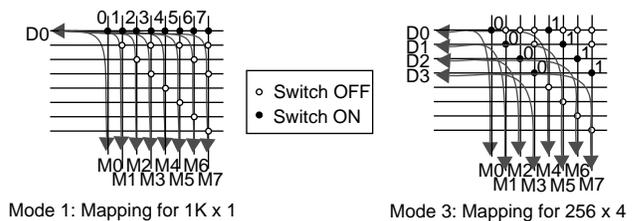


Figure 4. Two of the Four Different Modes of the Mapping Block

similar number of wires for two reasons: First, since individual wires in the address and data buses are always connected or disconnected in the same way, only a single configuration bit is needed for programming a connection from one bus to another, whereas in an FPGA the number of configuration bits is equal to the number of wires. This represents a 10-fold saving in the number of address bus routing configuration bits for the 10-line address bus, and similar savings on the data bus. The second saving arises because complete flexibility isn't required in routing structure. Although we chose to implement a routing structure that gives the maximum number of logical memory configurations possible with the four BMBs described above, this does not require all possible connections between all physical memory buses. By restricting certain logical memories to particular I/O pads a much simpler global routing architecture is possible, as illustrated in Figure 5 [Ngai94].

The solid circles in Figure 5 represent those switches that are closed when implementing a 4Kx1 logical memory. In this example, the address buses from all BMBs are connected together, as are all the data buses. Figure 5 also

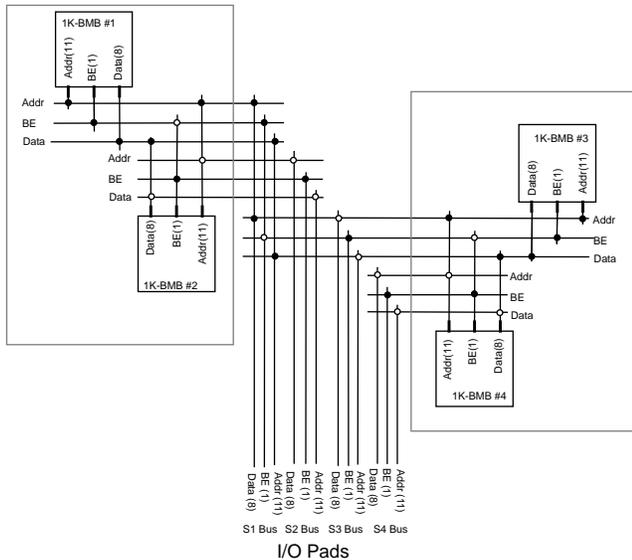


Figure 5. Routing Architecture and Setting for 4Kx1

illustrates the Block Enable lines, which are used to select the appropriate BMB when more than one BMB is used to form a logical memory.

The total flexibility of an FCM is determined by the combined flexibility of its BMBs and the routing structure that groups the BMBs together. In total, 133 different configurations are possible, counting only the logical memory configurations which utilize all 4Kb of memory. Table 1 gives a partial listing of the 133 logical memory

configurations, organized by the number of distinct logical memories.

One Logical Memory	Two Logical Memories	Three Logical Memories	Four Logical Memories
4K x 1	TWO 128 x 16	TWO 1K x 1 and 256 x 8	FOUR 256 x 4
1K x 4	1K x 1 and 128 x 24	1K x 2, 256 x 4 and 128 x 8	TWO 1K x 1 and TWO 256 x 4
256 x 16	1K x 2 and 256 x 8	512 x 2, 256 x 8 and 128 x 8	TWO 128 x 8, 512 x 4 and 256 x 4
128 x 32	512 x 2 and 384 x 8	256 x 4, 128 x 16 and 128 x 8	1K x 1, 512 x 2, 256 x 4 and 128 x 8

Table 1. Partial listing of the 133 possible logical memory configurations

In addition, a pseudo dual-port feature is incorporated to allow two BMBs be grouped together to support dual-port reading but single-port exclusive writing. This is achieved by dynamically switching the routing structure so that data can be written into two BMBs through the same bus and read back separately on two different external buses. The total number of logical memory configurations for FiRM including those employing this dual-port feature is 203.

3.3 Programming

FiRM is programmed using a 40-bit shift register chain that stores the configurations of the four BMBs and the routing structure.

4. Chip Status and Measured Performance

The FiRM chip has been fabricated using a 1.2 μm double-metal CMOS technology. Figure 6 illustrates its floorplan and dimensions. This design is pad limited, requiring 14.70 mm^2 for the inner core and 7.08 mm^2 for the active area. The white space inside the pad ring was used to fabricate a separately bondable instance of the base 128x8 memory in order to measure the speed of the memory array, and so estimate the speed penalty due to the Field-Configurability. Figure 7 shows a microphotograph of the chip.

The tested chip is partially functional: the separately bonded memory works correctly, and the programmable address and data bus connections also function. Although some memory locations of the FiRM can be read and written reliably with particular test vectors, other memory accesses do not work correctly. At this point in the testing, we are unable to determine the cause of the problem. We are, however, able to measure the access time for the working test vectors.

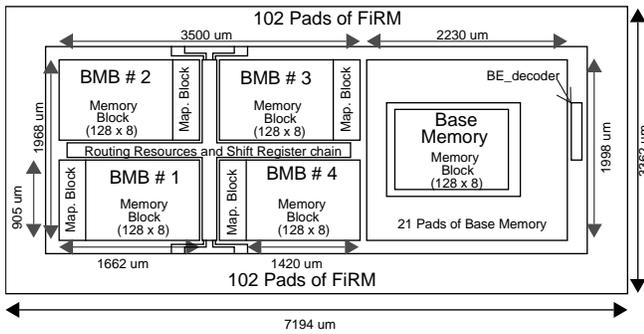


Figure 6. Dimensions of the FiRM Chip and Base Memory Chip

Figure 7. Chip microphotograph.

Table 2 gives the area measurements of the base memory and of the FiRM chip. The mapping block only occupies 10% of the area of the complete BMB. The FiRM core area is only 38% larger than the area for four base memory arrays without the configurability. The difference is due to the programmable routing structure, the mapping blocks and the programming circuitry. Note that this difference of 38% compares very favorably to the ratio between FPGA logic density and Mask-Programmable Gate Array logic density, which is roughly one to ten. Compared to the Xilinx 4000 series internal memory, after normalizing for process differences [Ngai94], the FiRM chip has from 16 to 23 times greater memory density.

The clock read access time (from rising edge of the clock until the data is valid) of the FiRM chip was measured at 40ns (under the tester load). The read access time for the separately bonded 128x8 memory was measured at 28ns using the same load. Therefore, the FiRM chip is 46% slower than the ASIC memory that it is based on.

5. Conclusions

In this paper, we have described the design of a full-custom field-configurable memory that combines the speed and density of the traditional SRAM design and the flexible structure of the FPGA.

Acknowledgments

The authors would like to thank CMC for fabricating the chip, Jaro Pristupa for his technical support, and Ken Schultz for advice and help on the memory design.

Area of BMB	1.50 mm ²
Area of Mapping Block	0.156 mm ²
Flexibility Overhead of BMB	10%
Active Area of FiRM	7.08 mm ²
Area for Four 128x8bit Memory Blocks	5.14 mm ²
% diff. of FiRM and 4 Memory Blocks	38%
Total die area of FiRM	24.2 mm ²

Table 2. Area Characteristics of the FiRM

References

- [Brit94] Barry K. Britton, Dwight D. Hill, William Oswald, Nam-Sung Woo and Satwant Singh, "Optimized Reconfigurable Cell Array Architecture for High-Performance Field-Programmable Gate Arrays", Proc. CICC '93, March 1993, pp. 7.2.1- 7.2.5.
- [Hsie90] H. Hsieh, K. Duong, J.Ja, R. Kanazawa, L. Ngo, L. Tinkey, W. Carter and R. Freeman, "Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays", Proc. CICC '90, May 1990, pp. 31.2.1 - 31.2.7.
- [Marp92] David Marple and Larry Cooke, "An MPGA Compatible FPGA Architecture", Proc. CICC '92, May 1992, pp. 4.2.1 - 4.2.4.
- [Ngai94] T. Ngai, "An SRAM-programmable Field-Reconfigurable Memory," Master's thesis, University of Toronto, 1994.
- [Plus89] FPSL5110 Product Brief, Plus Logic Inc., San Jose, CA., Oct. 1989.
- [Smit93] Daniel E. Smith, "Intel's FLEXlogic FPGA Architecture", Compton Spring '93, Feb. 1993.
- [Wilt95] S. Wilton, J. Rose, and Z. Vranesic "Architecture of Centralized Field-Configurable Memory," to appear in ACM Int'l Symp on FPGAs, FPGA '95.