ACSP · Analog Circuits and Signal Processing

Baker Mohammad

# Embedded Memory Design for Multi-Core and Systems on Chip

Springer

# Analog Circuits and Signal Processing

*Series Editors*
Mohammed Ismail, The Ohio State University
Mohamad Sawan, École Polytechnique de Montréal

For further volumes:
http://www.springer.com/series/7381

Baker Mohammad

# Embedded Memory Design for Multi-Core and Systems on Chip

Springer

Baker Mohammad
Khalifa University of Science, Technology and Research
Abu Dhabi, United Arab Emirates

ISSN 1872-082X ISSN 2197-1854 (electronic)
ISBN 978-1-4614-8880-4 ISBN 978-1-4614-8881-1 (eBook)
DOI 10.1007/978-1-4614-8881-1
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013948915

© Springer Science+Business Media New York 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

Embedded memory plays a big role in digital systems applications due to the increase of the data size required by many of these applications, such as video games and communication protocols. Also, the ever-increasing gap between processor speed, main memory, and bus speed (memory wall) creates a need for more on-chip memory to keep the processor busy and increase throughput. In addition to the increase of processor frequency, the integration of many cores or functional units on the same chip, which is referred to as system on chip (SOC), requires larger memory size. Embedded memory compromises more than 50 % of the chip area and greater than 80 % of transistor counts. Increased process variation due to technology scaling and the desire for high density memory results in a big challenge to meet the stringent requirements on performance, power, and yield. Embedded memory does not only play a positive role in system performance, but it also has an impact on yield, timing, and power. Memory organization and early decision made by system level and architecture group have big influence on the role and the impact the memory has on the overall system. Tradeoffs from memory cell type, array organization, memory hierarchy, Design for Test, and overall memory subsystem have to be considered early on.

This book reflects the latest trends in memory design and build on incorporating the result of cutting edge research and build of real product during my over 16 years' experience in the field. It is expected to be used by researchers, engineers, and graduate students. The unique feature of the book is its breadth and depth of memory design in small geometry process technology from system level, RTL, verification, circuit design, and into Design for Test.

Abu Dhabi, United Arab Emirates — Baker Mohammad

# Contents

# List of Figures

# List of Tables

# Chapter 1
# Introduction

## 1.1 Embedded Memory Importance

Embedded memories are becoming an increasingly important part of processor and system-on-chip (SOC) because of their positive impact on performance. However, embedded memories can negatively impact area, power, timing, yield, and design time. The ever-increasing gap between processor frequencies and DRAM access times, popularly referred to as memory wall, has implicated that processors use more and more on-die memory, hence the name "Embedded memory" [1, 2]. In addition, the new paradigm of multi-core systems and multi-functional units on the same die driven by the need for power efficiency, multi-functioning and large data size for high performance also contributes to the increase of embedded memory size [3]. As a result, in many chips the memory arrays make-up more than 80 % of the device and occupy about half of the chip's area [4]. Figure 1.1 shows an example of the embedded memory size trend of the Intel mobile processor [5].

Process scaling, with the ability to double the number of transistors in each generation (Moore's law) of technology, ultimately makes it possible to double the number of cores (processor unit) of each generation. DRAM has been the preferred choice for off-chip main memory and its primary emphasis on density rather than speed has increased the performance gap between the processor unit and the main memory [3].

Memory subsystem design and hierarchy are important aspects of the overall system performance, power and size, and a close attention needs to be paid to achieve the overall system goal. This goal of the memory subsystem is to provide the execution unit with the needed data and instructions as fast as possible and with smallest timing and power overhead. The processors ideally like to see infinite memory size and zero access time to memory; hence the memory hierarchy and design goal is to come as close to this ideal condition as possible. This requires low miss rate and short memory access time relative to the processor. An efficient memory subsystem tries to hide latency and minimize the power by implementing memory hierarchy [6, 7]. The tradeoffs between memory capacity, cell type, cell size, access time and power all need to be considered early on in the design phase in order to

B. Mohammad, *Embedded Memory Design for Multi-Core and Systems on Chip*,
Analog Circuits and Signal Processing 116, DOI 10.1007/978-1-4614-8881-1_1,
© Springer Science+Business Media New York 2014

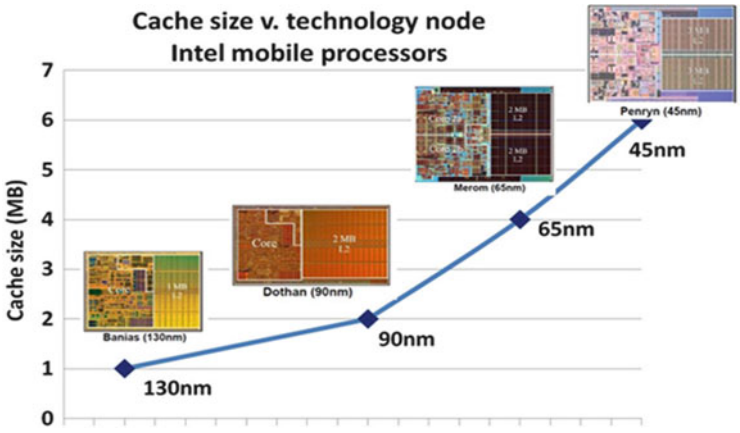


**Fig. 1.1** Intel mobile processor and embedded memory size

achieve an optimal memory subsystem. The next section discusses the memory types available for on-chip memory subsystems.

## 1.2 Embedded Memory Types

One of the first decisions that needed to be made early on in the development of any embedded processor or SOC design with regard to memory is the type of memory needed. Most of the time, this is a combination of the available technology and cells type [11]. We will categorize the choices under three types:

### *1.2.1 Volatility*

The first choice to be made is whether we need a volatile or a non-volatile memory. For embedded memory the majority is volatile [2], but in some cases we need specialized memory that can retain its state even after the power is turned off (non-volatile). A good example of this type is a flash memory that is commonly used in many USB and mobile memory. For on-chip, the usage of non-volatile is limited to specialized program that is stored in programmable read-only-memory (ROM) [8]. The purpose is to store information and programs that are closely tied to the hardware such as processor ID, firmware, configuration register and critical lookup tables. They can be programmed at testing time. Also, as we will discuss in Chap. 9, there are some new emerging technologies that provide non-volatility with reasonable access time and low energy which must be considered to achieve low power use especially leakage power. The volatility is not used for long retention but rather as part of the overall power management scheme.

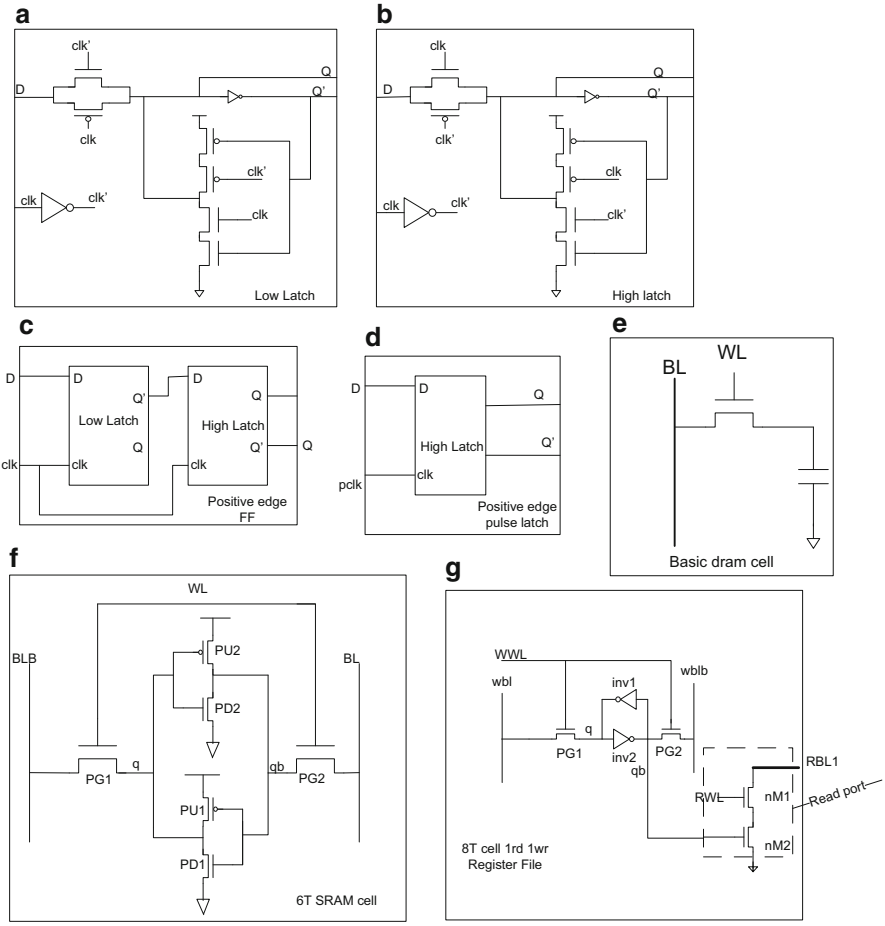


**Fig. 1.2** Main stream embedded memory cell type and their schematic: (**a**) low latch; (**b**) high latch; (**c**) positive edge trigger flip-flop consist of two latches (low → high), negative edge trigger will be the same except (high → low); (**d**) pulse latch; (**e**) DRAM cell; (**f**) 6T SRAM; (**g**) register file cell with multi-port (one read and one write)

The second type of memory is a volatile memory where the memory keeps its value as long as power provided to the system. Examples are SRAM, DRAM, latches, and flip-flops. Figure 1.2 illustrates the schematic view of the different cell types.

### *1.2.2 Memory Cell Type*

Memory cell type is dependent on the process technology used to produce it and has a big impact on all important aspects of its design metrics. The non-volatile memory cells available for consideration for each memory type will mainly depend on size

(kilo bits), speed (access time in seconds), and on the number of ports of the cell. Some specialized functionality also may be needed like comparison in the case for content addressable memory (CAM) [9]. Collaboration between architecture, circuit, and technology engineers is required to select the cells that can achieve the desired size and speed with the smallest area, lowest energy consumption, and highest product yield. The most popular cell types can be categorized as follows:

1. Flip-flop or latch-based: This type is mainly used to store a small set of data that need to be accessed frequently and quickly. It is the type of storage that is normally used to separate one pipe stage from the next one. Small FIFO size used for synchronization can also utilize these flip-flops or latches. Currently, many companies utilize flop trays where multiple flops are designed as one cell to provide a single register. The advantage of this kind of design is that it provides a relatively smaller area and smaller clock through the utilization of a 16- or 32-flop layout instead of 16 or 32 individual cells [10]. Flip-flops and latches are normally provided as standard cells for each technology nodes and synthesis tools can easily place and route them. A typical microprocessor has 10–15 % of its area consumed by sequential elements.
2. Register file-based: This is used for multi-port memory where one state element can be accessed by multiple execution units at the same cycle. The typical size of this storage area is 1–2 KB and the number of ports can vary from 4 to 12 read and 4 to 8 write ports [12, 13]. Figure 1.2 shows a one read and one write register cell. The addition of a write port is possible by adding two transistors per port parallel to PG1 and PG2 with new WWL to control the access. Also adding a read port is possible by duplicating the read port nM1 and nM2 transistors and connecting new RWL to nM1. In most processors, the access time to this register file is critical and needs to be as fast as possible as it is the source of all the operands of the execution units. A phase-based design is employed where one clock phase is used for read and another is for write.
3. SRAM: This is often referred to as a 6T cell because it uses six transistors: four for storage and two for access. Due to the symmetry and layout regularity of the cell, it uses small transistor sizes which are even narrower than the minimum width required by the target process technology. Its small size enables it to have a high memory density; the smaller the cell size, the more memory can be established in a given area. The advantage of SRAM cells is their high speed and smaller sizes compared to all previously mentioned cells, but their design complexity and sensitivity to process variations are some of the challenges that need to be addressed when using SRAMs [14, 16, 17]. All foundries produce multiple SRAM cells to tradeoffs between area, leakage power, performance, and yield. For example in 45 nm there are three variations of SRAM cell with cell size in $\mu^2$ (0.299 high density, 0.342 high performance, and 0.374 low voltage). Traditionally, 6T is used for all memories greater than 1 KB; this includes caches level 1, level 2, etc. Chapter 4 will discuss more details about SRAM-based design and cell selection.

4. CAM: In many cases, register data content like an address needs to be matched with many other entries in a memory. This requires a memory system to have the ability to make distinctions in the input register among all the entries of the memory. For example, a translation look aside buffer (TLB) where the virtual page number needs to be searched through in a storage area with multiple entries to ascertain if the page is in the TLB or not. This kind of memory requirement is best implemented with a CAM cell [7, 9]. There are many topologies of CAM cells and which to select is based on design complexity and timing requirement. Some caches tag arrays are also implemented with a CAM approach which we will discuss in more detail in Sect. 2.6.
5. Embedded DRAM: Dynamic RAM is often referred to as 1T1C as the cell is made up of one transistor and one capacitor. The nMOS transistor is used to control access and the capacitor is used to store charge. There is a new trend for multi-core processors to use level 3 caches in excess of giga byte size [18]. In this type of processor architecture, an embedded dram can be a more economical option than SRAM. The tradeoff is the added costs for the DRAM masks versus saving in area from using smaller DRAM than SRAM cell. The challenge of DRAMs is their need for refreshing cycle and their relatively slower access time compared to SRAMs. However, for higher-level caches a higher latency can be tolerated.
6. Non-volatile memory (FeRAM, PCRAM, STTRAM): The use of some types of emerging technology especially non-volatile memory like the ones listed above has been exploited for the development of on-chip memory especially for embedded systems (e.g., automotive, aerospatial) that require non-volatility to store code [8, 20]. However, main stream processors and SOC have not adopted these new types of technologies due to FeRAM scaling issues, PCRAM high voltage and temperature sensitivity, and STTRAM read disturb. We will discuss these technologies and their potentials in Chap. 9.

## 1.3 Memory Implementation with Discrete Component

There are two approaches to constructing a memory system; the first is to use discrete components such as flip-flops (FFs), level-sensitive latches (LSLs), and pulse latches (PLs) in addition to multiplexer and other combinational circuits. The second approach is to construct an array utilizing specialized cells like SRAM, and register file cells. The first approach is well suited to small to medium size arrays (FIFO, state arrays, and small register files) where area and power may be traded off for a simple design. This type of design requires discrete multiplexing of output data. Automation and CAD tools using ASIC design methodology can also be used to implement this first approach. Figure 1.3 shows the main component of this type of memory with its discrete components. In addition to the multiplexer on the output data, the clock for each row is gated using address bits for the memory data during the write operation.

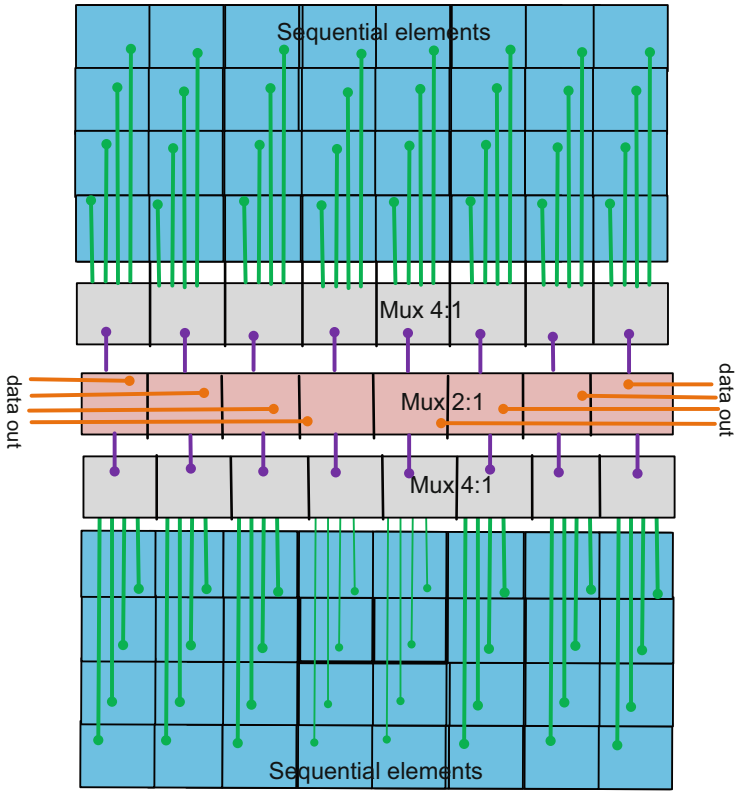


**Fig. 1.3** Memory implementation using sequential element (FF, pulse latches)

A comparison between the three sequential elements (FF, LSL, PL) in terms of area overhead, timing overhead, clocking, power, and design complexity will be discussed because these metrics need to be considered when deciding which sequential element to use. Flip-flop has the highest area overhead because it consists of two latches master and slave, followed by pulse latch [2]. All sequential elements require setup and hold time during write operation. Setup time is referred to the time required for the input signal to be stable for the sequential element to correctly store it. It is needed due to internal sequential element access delay and latching element. Hold time is needed for the signal to be stable after the capture clock closes the sequential element. On the read operation clock to out delay plus all the multiplexing is part of the access time overhead.

Figure 1.4 illustrates the main difference in timing between the three discrete elements: flop, level-sensitive latch, and pulse latch. Since this book discusses array design, LSLs are not suitable because they put limitation on both the write and read logic to use latches as well.