Digital Integrated Circuits – A Design Perspective 2/e

Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolić

Chapters 6 and 12

Design Project: 512-Word Content-Addressable Memory

1. Background

Content Addressable Memories (CAMs) are used in a variety of applications requiring pattern matching operations on bits, such as virtual memory, data compression, caching, and table lookup applications. With the popularity of wireless networking on the rise, CAMs have been suggested as a method of network-address filtering and translation by matching partial node addresses. In TCP/IP parlance, this is equivalent to matching the relevant bits of an IP address, as determined by the subnet mask. Clearly, in a wireless environment, power dissipation is of principle importance. However, base station nodes can afford to spend more power to achieve fast table lookups. Since cost is always a factor in commercial designs, minimizing the area of the chip is also important.

In this semester's project we will design a critical part of a CAM, under different design constraints.

1.1. High level structure

The high-level block diagram of a CAM is shown in Figure 1.

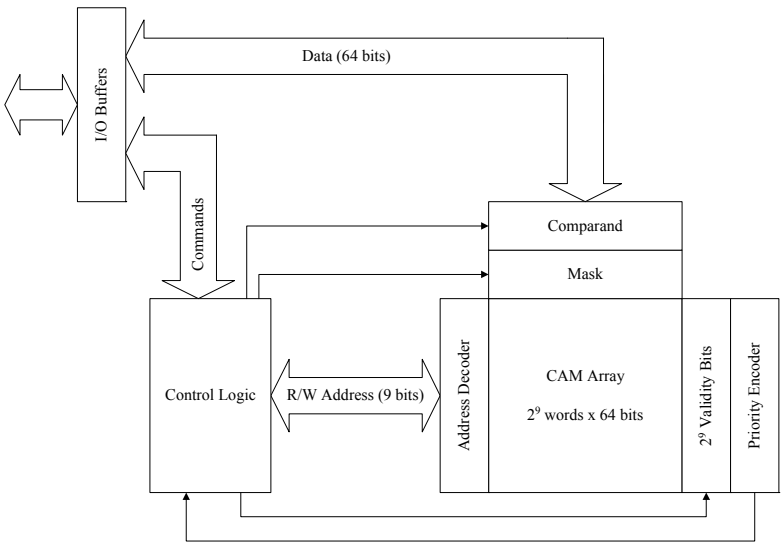

Figure 1. CAM high-level block diagram

CAMs have three modes of operation: read, write, and match. The read and write modes access and manipulate data in the CAM array in the same way as an ordinary memory. The benefit of

the CAM is realized in the match mode of operation. In this mode, the comparand block is filled with the data pattern to match, and the mask block indicates which bits are significant. For example, to find all the words in the CAM array that have the pattern 0x123 in the most significant bits, we would fill the comparand with 0x12300000 and the mask with 0xFFF00000. All 512 rows of the CAM array would then simultaneously compare the twelve most significant bits of the comparand with the data contained in that row. Every row that matches the pattern is then passed to the validity block. Since we do not care about rows that contain invalid data (which typically happens when the array is not full), only the valid rows that match are passed to the priority encoder. In the event that two or more rows match the pattern, the address of the row in the CAM array is used to break the tie. To do this, the priority encoder considers all 512 match lines from the CAM array, selects the one with the highest address, and encodes it in binary. Since there are 512 rows in the CAM array, 9 bits will be required to indicate the highest row that matched. One additional 'match found' bit will be required, since it is possible that none of the rows matched the pattern.

As a simple example, consider the CAM array when filled with the six entries shown below in Table 1.

Table 1. Example CAM array with six entries

Address	Data
0	0x674243A4
1	0x6725643E
2	0x68425786
3	0x12D64368
4	0xE4B64802
5	0x75367843

Some example comparand/mask combinations and the results of the match operation are shown in Table 2.

Table 2. Example match operations

Comparand	Mask	Match?	Address
0x674243A4	0xFFFFFFFF	Y	0
0x89424858	0x00FF0000	Y	2
0x05943864	0x0000F000	N	N/A
0x3859AB34	0x00000000	Y	5

2. Implementation and Constraints

The goal is to design the priority encoder to be used in a CAM with a particular set of optimization criteria. The project will be completed in THREE phases.

The goal of the first phase is to perform the logic optimization, circuit style selection, and first-order COMBINATIONAL circuit optimization to meet the stated design goals and constraints.

The fine-tuning of the design and the actual physical layout of the priority encoder will be performed in PHASE 2.

You should select one of the following design cases:

a) Low energy operation: Design a priority encoder to be used in a CAM in a wireless node, such that the average energy is minimized while still meeting the constraint that the worst-case delay is smaller than 40 ns. No constraints are put on the area of the design.

b) High speed operation: Maximize the speed of the priority encoder for the case where the CAM is to be used in a basestation and power is readily available. No constraints are put on area or power.

c) Low silicon cost: Minimize the area of the priority encoder, while meeting the constraint that the worst-case delay is smaller than 40 ns. No constraints are put on energy consumption.

The project is to be done in pairs.

You are free to choose any logic family for the impolementation of the project: complementary CMOS, pseudo-NMOS, pass-transistor logic, dynamic logic, etc.

2.1. TECHNOLOGY: The design is to be implemented in a 0.25 μm CMOS process with 4 metal layers. The SPICE technology is in the g25.mod file.

2.2. POWER SUPPLY: You are free to choose any supply voltage and logic swing up to 2.5 V. Make sure that you use the appropriate model when you perform hand analysis.

2.3. PERFORMANCE METRIC: The propagation delays for static designs is defined as the time interval between the 50% transition point of the inputs and the 50% point of the worst-case output signal. Make sure you pick the worst-case condition and state EXPLICITLY in your report what that condition is. Note that for dynamic designs the propagation delay is defined in this case as the delays of the evaluate phase ONLY (at least in this phase of the project)!

2.4. AREA: The area is defined as the smallest rectangular box that can be drawn around the design. Note that since the priority encoder must be interface with the validity block and the CAM array, all of the row-match inputs must be accessible from the left side of the design, in row-address order. In the first phase of the project, you should make the area estimations based on the total transistor width and the wiring complexity. An expression on how to predict the area will be provided shortly on the web-page.

2.5. NAMING CONVENTIONS: You should label the inputs and the outputs of the design as it is shown in Figure 2. The valid-match signals are denoted as *vm[0]* to *vm[511]* where the index is the address of the row. The encoded-match signals are denoted as *em[0]* to *em[8]*, where *em[0]* is the least significant bit. The *match* signal is high when the encoded-match signals indicate a valid match and low otherwise.

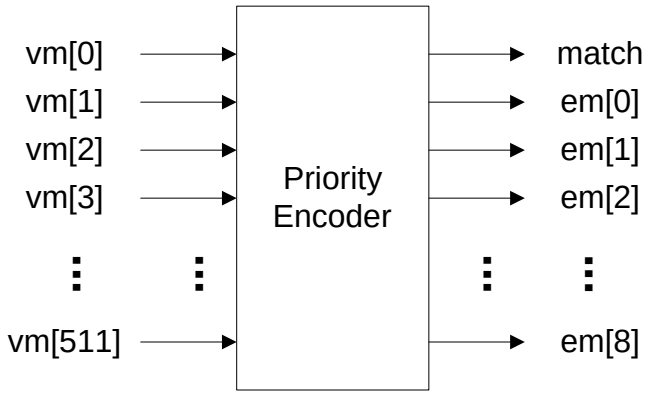

Figure 2. Signal naming conventions for priority encoder

2.6. REGISTERS: In this phase of the design you do not need registers. The data flow from input to output should be combinational logic.

2.7. CLOCKS: There should be no global clock, since the design is combinational. If you choose dynamic logic, you are permitted a precharge/evaluate clock, but the result must become available after ONE evaluate stage (no pipelined logic). Observe that the load capacitance of the clock should be included in the energy analysis.

2.8. V_{OH}, V_{OL}, NOISE MARGINS: You are free to choose your logic swing. The noise margins should be at least 10% of the voltage swing. Test this by computing the VTC between one of the inputs and the output signals (with the other outputs set to the appropriate values) for a static design. For a dynamic circuit, apply an input signal with a 10% noise value added to the input and observe the outputs.

2.9. RISE AND FALL TIMES: All input signals have rise and fall times of 500 ps. The rise and fall times of the output signals (10% to 90%) should not exceed 1 ns.

2.10. LOAD CAPACITANCE: Each output bit of the priority encoder should have a load equivalent to 4 unit inverters (Wn = 0.25 μm, Wp = 0.50 μm).

3. Simulation

Analyze the circuit by using either SPICE or IRSIM to simulate the design. Prove the functional operation of your circuit using either SPICE or IRSIM. The input patterns to be used to determine functionality and energy dissipation will be provided on the web-page. The pattern that causes the worst-case propagation delay should be determined by yourself, based on an analysis of your circuit schematic. Make sure that you define your circuit in a hierarchical fashion to

4. Report

The quality of your report is as important as the quality of your design. One must sell the design by justifying the design decisions and by providing all the vital information. Be sure to

emphasize relevant information and to eliminate unnecessary material. **Organization, conciseness, and completeness are of paramount importance.** Do not repeat information we already know. Use the templates provided on the web-page (in Framemaker, Word, and PDF formats). E-mail your electronic submission of your report as a Framemaker, Word, or PDF file. Make sure to fill in the cover-page and use the correct units. A report has to be submitted at the end of each phase of the project.

4.1 Report Contents

Your report should discuss your overall design philosophy and the important design decisions made at the logic and circuit level. Discuss why your approach increases the operating speed or helps to reduce energy or area, while meeting the performance specifications. Provide your current estimates of the results and describe how you arrived at them. Include schematics and highlight the important elements.

Prove that your results are accurate by providing the crucial plots (don't forget to mention the input patterns used to obtain those plots). The total report should not contain more than three pages. You are not allowed to add any other sheets, except for important plots. The organization of the report should be based on the following outline:

Page 1: Executive summary, overall design decisions, remarks, and motivations
Page 2: Logic and transistor diagrams, annotated with transistor sizes and worst-case timing path. Plot showing the functional operation of the cell. Comments.
Page 3: Timing and energy simulations – derive the value of the worst-case path and average energy. For the latter, a set of test patterns will be provided on the web page.

Lastly, you are required to submit the SPICE INPUT DECK by e-mail. Remember, a good report is like a good layout: it should perform its function (convey information) in the smallest possible area with the least delay and energy (to the reader) possible.

The quality of the report is an important (major) part of the grade!

The total project grade is divided into the three phases

40% Phase 1
20% Phase 2
40% Phase 3

For each phase, the grade will be divided as follows:

30% Results
20% Approach and correctness
40% Report
10% Creativity

