International Journal on Applications of Information and Communication Engineering
Volume 1: Issue 2: February 2015 , Pages:45-49 www.aetsjournal.com

Design and Implementation of Content Addressable Memory for System on Chip

R. Reshma, S. Nikilla

***Abstract*—**Content addressable memory (CAM) is a special type of computer memory used in certain high speed searching applications. CAM compares input search data against a table of stored data and returns the address of the matched data. CAMs can perform its searching application in a single clock cycle. The main challenge in designing the CAM is to reduce the power consumption without reducing the speed and memory density of the search data. In this paper, we introduce anoptimised parity based CAM which leads to delay reduction, area and power overhead. The parity CAM can then be applied to the proposed ARM architecture that attempts to simultaneously reduce power with highest dissipation

Keywords— Content addressable memory(CAM), match-line, search-line, System on chip, ARM processor.

I. INTRODUCTION

Most memory devices store and retrieve data by addressing specific memory locations. For example, a system using RAM or ROM searches sequentially through memory to locate data. However this technique reduces the performance of memory since the searching of data requires multiple clock cycles [1]. CAM considerably reduces the time required to search by identifying stored data by content rather than by address. Memory accessed in this way is called Content addressable memory (CAM).

A. Content Addressable Memory

CAM offers a performance advantage over other memory search algorithms [2], such as binary based searches, tree based searches or look aside tag buffers, because it compares the desired information against the entire list of pre- sorted entries simultaneously. Thus, CAM provides an order of magnitude reduction in the search time. CAM is based on RAM technology [1].

Content addressable memory (CAM) is a solid state memory in which the data are accessed by their contents rather than physical location [3,4]. CAM has simple storage cells; each individual memory bit in a fully parallel CAM must have its own associated comparison circuit to detect a match between stored bit and input bit. The match output from each CAM cell in the data word can be combined to produce a complete data word match signal.CAM is almost faster than RAM [5,6].CAM can offer high speed search function in a single clock cycle [7,8].

Generally CAM has three operation modes: READ, WRITE and COMPARE in which compare is the main operation [10, 11, and 12]. The CAM can be applied to various applications and the approach proposed in this paper aims to reduce the power in register file, instruction queue, reorder buffer, load/store queue.

The concept of the parity CAM is described in section II, and followed by section III, which introduces the proposed CAM cell design. Section IV describes the proposed ARM architecture that is the processor units. Comparison results and the conclusion are presented in section V and VI.

B. System on Chip

An SOC is a system on an IC that integrates hardware and software Intellectual property (IP) using more than one design methodology for the purpose of designing the functionality and behaviour of the proposed system. SOC designs include embedded processor cores, and thus a significant component, which leads to additional methodology, process, and organizational challenges.

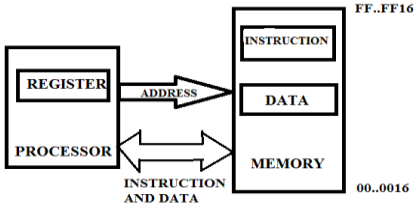

Fig 1. Processor Architecture

A general purpose processor is a finite state automation that executes instructions held in a memory. The state of the system is defined by the values held in the memory locations together with the values held in certain registers within the processor itself. Each instruction defines a particular way the

R. Reshma is with the Department of Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, Tamilnadu, India, (e-mail: reshma3512@gmail.com)

S. Nikilla. is Assistant Professor, Department of Electronics and Communiction Engineering, Sri Krishna College of Engineering and Technology, Tamilnadi, India, (e-mail: nikilas@skcet.ac.in)

total state should change and it also defines which instruction should be executed next.

The ARM processor design is based on RISC principles.. Its code density is still, however, not as good as some CISC processor. Where code density is of prime importance, ARM limited has incorporated a novel mechanism, called the Thumb architecture, into some versions of the ARM processor. Then ARM processor is at the centre for power efficient processing. It therefore seems appropriate to consider the issues around design for low power.No processor is particularly useful without the support of hardware and software development tools. The ARM is supported by a toolkit which includes an instruction set emulator for hardware modelling and software testing and benchmarking, an assembler, C and C++ compilers, a linker and a symbolic debugger.

The organization of the ARM integer processor core changed very little from the first 3 micron devices developed at Acorn computers between 1983 and 1985 to ARM7 and ARM9 developed by ARM limited between 1990 and 1995. A 3-stage ARM core processor accesses memory on almost every clock cycles either to fetch an instruction or to transfer data. To get an significantly better CPI the memory system must deliver more than one value in each clock cycle either by delivering more than 32 bit per cycle from a single memory or by having separate memories for instructions and data accesses. As a result of the above issues, higher performance ARM cores employ a 5-stage pipeline and have separate instruction and data memories. Breaking instruction execution down into five components rather than three reduces the maximum work which must be completed in a clock cycle, and hence allows a higher clock frequency to be used. The separate instruction and data memories allow a significant reduction in the core's CPI.

II. Proposed Parity Content Addressable Memory

The Parity bit based CAM design is shown in Fig. 2. It consists of the original data segment and an extra one bit segment. The Parity bit can be obtained based on odd and even number of 1's. Hence, with the help of Parity bit data can be accessed faster. For the even number of 1's the entry is '0' and for odd number of 1's the entry will be '1'. The new CAM design has the same interface as the conventional CAM with one extra bit.

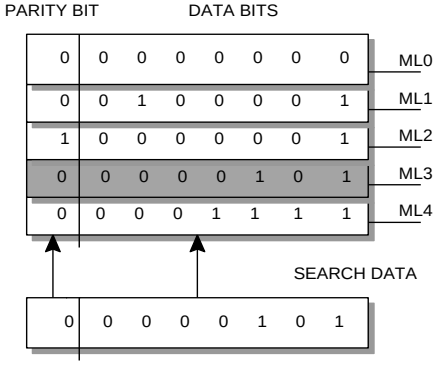

Fig. 2 Proposed Parity Cam

Hence, this Parity bit based CAM does not improve the power performance. Theoretically, this additional Parity bit reduces the sensing delay and boosts the searching speed. The Parity bit of the search and the stored word is the same in the case of ML3, thus the overall word returns a match. For example, in case of ML2 there is a mismatch between the stored data and the search data and so the number of 1's between the stored and search data must be different by 1. As a result, the corresponding parity bits are different. If suppose there are two mismatches in the data segment that is, in the case of Fig. 3 ML0, ML1 and ML4, the parity bits are same and therefore overall we have two mismatches. Now with the help of sense amplifier we have to identify the mismatched and the matched cases. This clearly shows that the driving strength of the 2- mismatch word is twice as strong as that of the 1-mismatch word, hence, the proposed Parity bit based CAM greatly improves the searching speed and the I_{on}/I_{off} ratio of the design.

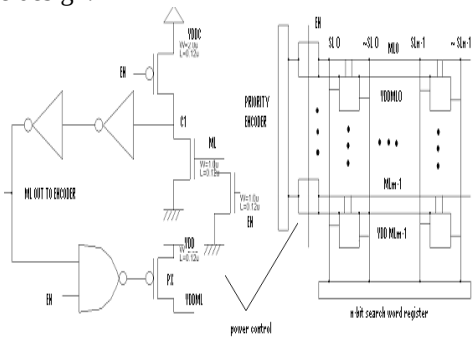

Fig. 3 Proposed gated ML design

III. Effective Gated Power Technique ML Design

The proposed CAM architecture is shown in the Fig. 3. In this design, the CAM cells are arranged into rows and columns. The row is indicated as a word and the column is indicated as a bit.

From the above Fig. 3, it is clearly shown that the COMPARISON unit, that is, transistors M1-M4 and the SRAM unit that is, the cross coupled inverters are shown by two separate metal rails VDD_{ML} and VDD. The purpose of having VDD_{ML} and VDD can fully isolate the SRAM cell from any power disturbances during COMPARE cycle.

In Fig. 3, the gated power transistor P_X, which is controlled by a feedback loop, denoted as Power Control can automatically turn off P_X, when the voltage on the ML reaches a certain threshold value. Now at the beginning of each cycle, the ML can be initialized by a global control signal EN. When the signal EN is set low and the P_x is turned OFF, this makes the signal ML and C1 to be held at ground and VDD respectively. Similarly, when the signal EN is set high, this initiates the COMPARE phase. When there are more mismatches in the CAM cells, the ML will be charged high. Whatever the number of mismatches may be, the current offered by the power transistor P_X will be limited.

IV. Design and Implementation of Processor Unit with I/O Interfacing

A. Processor Design

The art of processor design is to define an instruction set that supports the functions that are useful to the programmer while allowing an implementation that is as efficient as possible. Preferably, the same instruction set should also allow future, more sophisticated implementation to be equally efficient. The semantic gap between a high level language construct and a machine instruction is bridged by a compiler, which is a computer program that translates a high level language program into a sequence of machine instructions. Therefore the processor designer should define the instruction set to be a good compiler target rather than something that the programmer will use directly to solve problem by hand.

Several techniques have been proposed to reduce power. One such technique is the Processor units which consist of Register file, Instruction queue, Reorder buffer, Load/Store queue that attempts to simultaneously reduce power with highest dissipation.

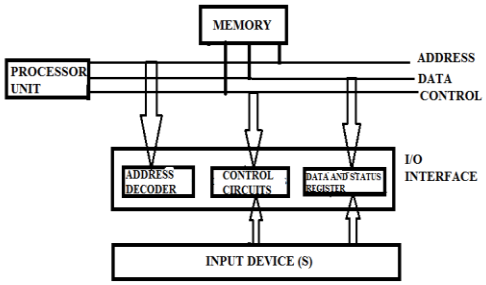

Fig 4. Processor unit with I/O interfacing

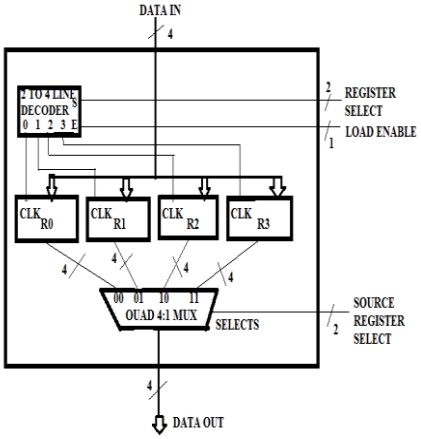

Fig 5. Register file

Most modern computers use single bus arrangement for connecting I/O devices to CPU and Memory. The bus enables all the devices connected to it to exchange information. Bus consists of three sets of lines that is Address, data and control lines.

- Processor places a particular address (unique for an I/O device) on address lines
- The device which recognizes this address responds to the commands issued on control lines
- Processor unit requests for either Read/Write
- The data will be placed on the data lines

a. Register File

The register file is capable of fetching, decoding, renaming several instructions per processor clock cycle. Later the processor can also execute and commit up to as many instructions as possible. This type of multiple user processor access the register file very frequently. The register file is one of the most active components in processor units because it is typically designed as an SRAM structure with many reads and writes ports as the maximum number of instructions the processor can issue in each cycle.

b. Instruction Queue

The instruction queue is a CAM+RAM structure that holds the instruction until they can be issued. The decoder decodes the instruction written into buffer A as soon as it arrives. The buffers B and C receive the decoded version of the instruction in buffer A.

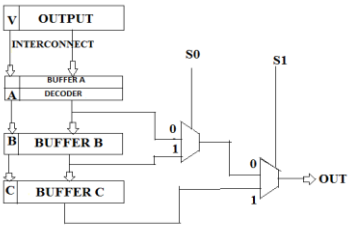

Fig 6. Instruction queue

The instruction queue first sets the entry for the new instruction. Once it sets the entry, it selects that entry to issue instruction to the functional unit. Later it wakeup the instruction which is waiting in the queue once the result is ready.

c. Reorder Buffer

Reorder buffer is a circular queue with head and tail pointers. It is a multiport SRAM structure with many functions.At the time of renaming, an instruction is assigned an entry at the tail of the reorder buffer which becomes the name of the result register. The instructions are issued in program order, thus the ROB stores the instruction in the program order.

Fig 7. Reorder Buffer

At the end of functional unit computation value is put in the instruction reorder buffer position. When the instruction reaches the head of the buffer, its value is stored in the logical or physical register. Reorder buffer entry contains type of instruction, destination and the value and its presence or absence. The reorder buffer, from its name it implies, can be used for in- order completion.

d. Load/Store Queue

The load/store unit is a CAM+RAM structure that is responsible for generating the virtual address of all load and store operations and also for accessing the data cache memory. One load or store operations can be issued per clock cycle. In order to further optimise the data store, two or more store instructions can be combined if they are in the same 8 byte block, so that a single data transfer occurs between the processor and the second level cache memory. Only load and store instructions access the memory whereas all other instruction uses register as operands. In load, source is memory and destination is register whereas in store source is register and destination is memory.

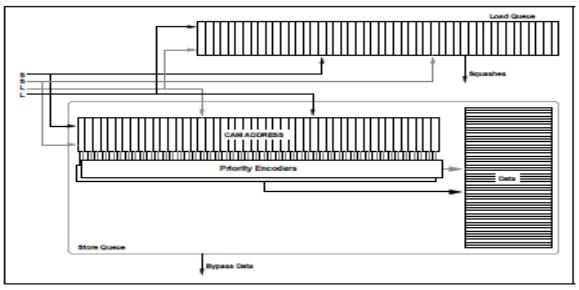

Fig 8. Load/Store Queue

Parallely they also perform an associative search of the other queue comparing address. If a store matches a load in a program order then a squash is signalled else if a load matches with two or more stores in a program order the index of the youngest is selected using a priority encoder and it also used to drive the RAM array that holds the stored value.

B. I/O System

The ARM handles I/O (input/output) peripherals as memory mapped devices with interrupt support. The internal registers in these devices appears as addressable locations within the ARM's memory map and may be read and written using the same (load-store) instructions as any other memory locations.

Peripherals may attract the processors attention by making an interrupt request using either the normal interrupt or the fast interrupt input. Both interrupt inputs are level sensitive and maskable. Normally most interrupt sources share the IRQ interrupt with just one or two time critical sources connected to the higher priority FIQ input. Some systems may include direct memory access(DMA) external to the processor to handle high bandwidth I/O traffic.

V. RESULT

The Simulation results of the two CAM cells with the processor units are simulated using the simulated tool Modelsim SE 6.3f and the functional verification is done using Xilinx. Table I shows the simulation result of conventional CAM cell and proposed CAM cells with the processor units. The simulation result shows that proposed CAM takes less power consumption than that of the other design.

The Simulation results of the two CAM cells with the processor units provided by I/O interfacing are simulated using the simulated tool Modelsim SE 6.3f and the functional verification is done using Xilinx.

TABLE I
COMPARISON OF THE PROCESSOR UNIT

PARAMETER	EXISTING METHOD (PRE COMPUTATION CAM)	**PROPOSED METHOD (PARITY CAM-PROCESSOR UNIT)**
TOTAL POWER	219mW	**199mW**
AREA (GATE COUNT)	1728	**1495**
INPUTS	15	**15**
CLOCKS	171	**150**
DYNAMIC POWER	83.14	**49.25**
DELAY	10.221ns	**4.629ns**

TABLE II
COMPARISON OF PROCESSOR UNIT WITH I/O INTERFACING

PARAMETER	EXISTING METHOD (PRE COMPUTATION CAM)	PROPOSED METHOD (PARITY CAM-PROCESSOR UNIT WITH I/O INTERFACING)
TOTAL POWER	345mW	331mW
AREA (GATE COUNT)	18,710	18,477
INPUTS	15	15
CLOCKS	297	282
DELAY	12.12ns	9.78ns
DYNAMIC POWER	96.28	62.65

Table II shows the simulation result of conventional CAM cell and proposed CAM cells with the processor units provided by I/O interfacing. The simulation result shows that

proposed CAM with I/O interfacing takes less power consumption than that of the other design.

VI. CONCLUSION

In this paper, the proposed CAM based on Parity bit and effective gated-power technique offers several advantages when compared to the conventional design. The proposed CAM reduces power and boosts the searching speed. It is therefore more stable than the conventional design while maintaining the low power consumption property. This feature confirms that the proposed design is more suitable for ultra-low power applications in CMOS process. The parity CAM is then applied to the proposed processor unit that attempts to simultaneously reduce power. The designed processor unit is optimized in terms of low power, minimum clock cycles for single operation, less latency search function, reduces delay for sensing address and robust against process variation.

REFERENCES

[1] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low power CMOS fully parallel content - Addressable memory macros*," IEEE J. Solid- State Circuits,* vol.36, no. 6, pp. 956- 968, jun. 2001.

[2] S. Baeg, "Low-power ternary content Addressable memory design using a Segmented match line," *IEEE Circuits Syst. I, Reg. Papers*, vol. 55,*Trans.* no. 6,pp. 1485-494, Jul. 2008.

[3] Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content addressable memories," *IEEE J. Solid State Circuits,* vol. 38, no. 11, pp. 1958- 1966, Nov. 2003.

[4] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey, *"IEEE J. Solid State Circuits,* vol. 41, no. 3, pp. 712- 727, Mar. 2006.

[5] N. Mohan, W. Fung, D. Wright, and M. Sash dev, "A low-power ternary CAM with positive-feedback match-line senseamplifiers*,"IEEE Trans. Circuits Syst.I, Reg. Papers*, vol. 56, no. 3, pp. 566- 573, Mar. 2009.

[6] N. Mohan and M. Sachdev, "Low power dual match-line content addressable memory," in Proc. *IEEE Int. Symp. Circuits Syst. (ISCAS),* vol. 2, 2004, pp. 633-636.

[7] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and Hierarchical search-lines for low power content- addressable memories," in Proc. *IEEE Custom Intergrated Circuits Conf*., 2003, pp. 383- 386.

[8] K. Pagiamtzis and A. Sheikholeslami, "A low power content-addressable memory (CAM) using Pipelined Hierarchical search scheme," *IEEE J. Solid- State Circuits*, vol. 39, no. 9, pp. 1512- 1519, Sep. 2004.

[9] SatoruHanzawa, Takeshi Sakata, Kazuhiko Kajigaya, RiichiroTakemura, Takayuki Kawahara, "A large scale and low power CAM architecture featuring a one hot spot block code for IP address in a network router," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 853-861, Apr. 2005.

[10] O. Tyschenko and A. Sheikholeslami, "Matchline sensing using matchline stability in content addressable memorys (CAM)," *IEEE J. Solid- State Circuits*, vol. 43, no. 9, pp. 1972- 1981, Sep. 2008.

[11] A.T. Do, S. S. Chen, Z. H. Kong, and K.S. Yeo, "A low power CAM with efficient power and delay trade-off," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS),* 2011, pp. 2573- 2576.

[12] C.S. Lin, J.C. Chang and B.D. Liu, "A low power pre computation based fully parallel content addressable memory," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 654-662, April 2003.

[13] Y.-L. Hsiao, D.-H.Wang, and C.-W. Jen, "Power modeling and low-power design of content addressable memories," in*Proc. IEEE Int.Symp. Circuits and Systems*, vol. 4, 2001, pp. 926–929.

[14] Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content Addressable memory (TCAM) based on 4T static storage and including a current- race sensing scheme," *IEEE J. Solid- State Circuits*, vol. 38, pp. 155–158, Jan. 2003.

[15] T. Matsuda and K. Matsuda, "A new protocol processing architecture for high-speed networks," in *Proc. IEEE GLOBECOM*, Nov. 1996, pp. 798–803.

[16] T.-B. Pei and C. Zukowski, "VLSI implementation of routing tables: tries and CAMs," *IEEE INFOCOM*, pp. 515–524, 1991.

[17] K. Choi, R. Soma, and M. Pedram, "Fine grained dynamic voltage and frequency scaling for precise energy and performance trade off based on the ratio of off-chip access to on-chip computation times," *IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst.*, vol. 24, no. 1, pp. 18–28, Jan. 2005.

[18] M. A. Ramirez, A. Cristal, A. Veidenbaum, L. Villa, and M. Valero, "A new pointer-based instruction queue design and its power-performanceevaluation," in *Proc. IEEE Int. Conf. Comput. Des. (ICCD)*, 2005, pp. 647–653.