HIGH PERFORMANCE DMA DATA TRANSFER FOR MULTI-BIT ERROR CORRECTION

ASAPU HARIKA[1], MR.A. SAI KUMAR GOUD[2], MR. PRADEEP KUMAR REDDY[3]
[1]PG Scholar,[2]Associate Professor,[3]Associate Professor
Dept. of ECE (VLSI),
CMR Institute of Technology, Kandlakoya (v), Hyderabad, TS,India.

ABSTRACT:

Scratchpad memories are alternatives to caches in real-time embedded processors. They provide better timing predictability and lower energy consumption.Applications in multimedia, video processing, speech processing, DSP applications and wireless communication require efficient memory design since on chip memory occupies more than 50% of the total chip area .This will typically reduce the energy consumption of the memory unit, because less area implies reduction in the total switched capacitance.Recently, interest has been focussed on having on chip scratch pad memory to reduce the power and improve performance.

SPM's are prone to soft errors like Single event upsets (SEUs) and Single event multiple upsets (SEMUs). For correcting soft errors, we use Error Correcting Codes (ECC) like single-errorcorrection double-error detection(SEC-DED) and SEC-DEDdouble-adjacent error correction(SEC-DED-DAEC) and parity duplication approach.These approaches are used only error correction for less number of corrections and thus increases speed.

The key idea is to provide a replica for software managed SPM enforcing the hardware managed cache to keep a copy of non-cacheable SPM lines.By inclusion of DMA,data can be transferred faster to or from scratchpad memory.

Keywords: Error Correcting codes,DMA,Duplication schemes,error detection.

INTRODUCTION:

Scratchpad memory (SPM), also known as scratchpad, scratchpad RAM or local store in computer terminology, is a high-speed internal memory used for temporary storage of calculations, data, and other work in progress. In reference to a microprocessor ("CPU"), scratchpad refers to aspecial high-speed memory circuit used to hold small items of data for rapid retrieval. It is similar to the usage and size of a scratchpad in life: a pad of paper for preliminary notes or sketches or writings, etc.

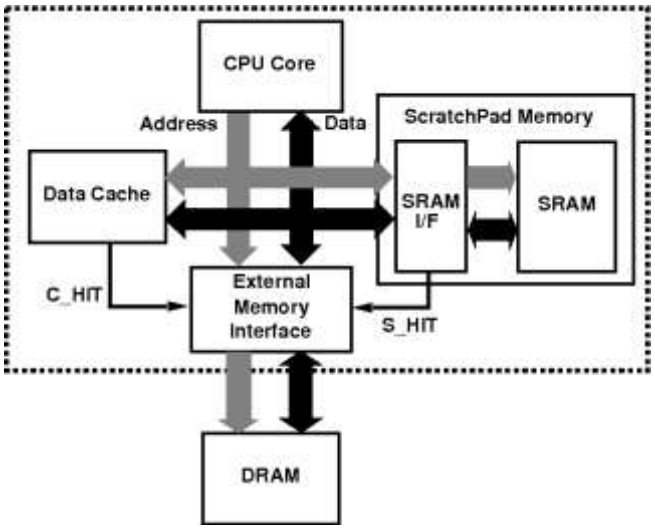

Fig 1. Scratchpad Memory

It can be considered similar to the L1 cache in that it is the next closest memory to the ALU after the processor registers.In contrast to a system that uses caches, a system with scratchpads is a system with Non-Uniform Memory Access latencies, because the memory access latencies to the different scratchpads and the main memory vary. Another difference from a system that employs caches is that a scratchpad commonly does not contain a copy of data that is also stored in the main memory.

They are mostly suited for storing temporary results (as it would be found in the CPU stack) that typically wouldn't need to always be committing to the main memory; however when fed by DMA, they can also be used in place of a cache for mirroring the state of slower main memory.Another difference is that scratchpads are explicitly manipulated by applications. They may be useful for realtime applications, where predictable timing is hindered by cache behaviour.

CADS:

The high vulnerability of SPM to soft errors, however, limits its usage in safety-critical applications. An efficient fault-tolerant scheme, called cache-assisted duplicated SPM (CADS), to protect SPM against soft errors. The main aim of CADS is to utilize cache memory to provide a replica for SPM lines. Using cache memory, CADS is able to guarantee full duplication of all SPM lines. We also further enhance theproposed scheme by presenting buffered CADS (BCADS) that significantly improves the CADS energy efficiency. BCADS is compared with two well-known duplication schemes as well as single-error correction scheme.

One major source of system failure in such applications is soft errors caused by radiation-induced particle strike into chips. Single-event upsets(SEUs) and single-event multiple upsets(SEMUs) are two types of soft errors in SPM and cache as ON-chip SRAM memories Correcting soft errors in ON-chip memories (SPM or cache) can be categorized into two approaches .

The first approach is the use of error-correcting codes (ECCs), e.g., single-error correction double-error detection (SEC-DED) and SEC-DED double-adjacent-error correction (SEC-DED-DAEC), to detect and correct errors. All of the ON-chip memories can be protected using this approach. However, this approach has two serious problems: 1) a limited error correction capability and 2) a significantly higher overhead, when ECCs are employed to correct multiple bit errors such as SEMUs. The second approach to detect and correct soft errors is a joint use of parity code and a duplication of memory entries; we call this approach parity duplication. The main advantage of this approach is its capability to correct all detected errors. In these structures, a copy of all entries is inherently available in lower memory levels and the overheads of memory protection mechanism are as low as the parity code overheads. However, the parity-duplication approach does not offer full protection for data-SPM and write-back data cache since a fraction of data blocks in these structures does not have any copy in the lower memory levels for error correction.The proposed cache controller provides the ability to allocate

and update the replica in the cache for SPM write accesses, to ignore the SPM read accesses, and to retrieve the replica for error recovery.

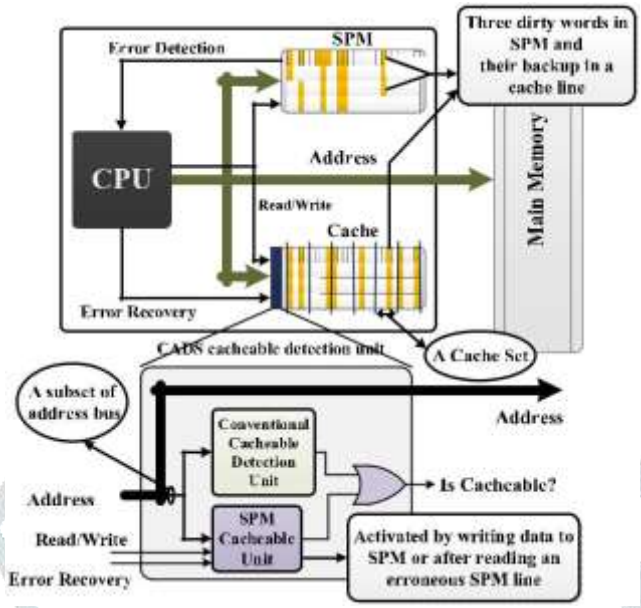

Fig 2.Cache-Assisted Duplicated SPM

CADS utilizes parity code to distinguish errors in SPM.Error Detection mechanism is likewise autonomously utilized from error recovery process. This can give two primary advantages: 1) the overheads for error correction are not forced in the ordinary system operation and 2) as indicated by the level of quality required for every application, a suitable error detection mechanism can be utilized, which makes this design appropriate to an extensive variety of applications. To distinguish both SEU and SEMU, interleaved parity code is considered for error detection, which has essentially lower overhead compared to ECC codes.

The error recovery process in SPM needs to separate among dirty and clean data lines. Errors happening in clean data lines can be rectified by its unique duplicate in the primary memory, while the duplicate of dirty information lines is accessible in the cache memory.This implies we propose two differentcaching policies in the cache for writing to SPM and reading from SPM.Each write access to SPM is characterized as a cacheable transaction in CADS; in this way, once an data line in SPM ends up dirty, a cache line is dispensed to keep a replica for this new dirty SPM line. In the ensuing updates of this data line in SPM, its reproduction in the cache is refreshed also. Then again, read task does not alter the SPM contents and no cache access is required. In this way, the ordinary caching policy is utilized for SPM read transactions.

To make the SPM write access as cacheable activities and to keep SPM read access as non-cacheable tasks, CADS upgrades the cacheable detection unit of the cache controller. The traditional cacheable detection unit chooses whether the address created by the processor is cacheable or not. In CADS, the read/write signal is additionally checked by the cacheable detection unit.The read/write signal demonstrates whether the write access and the address produced by the processor are in the SPM area, the cacheable detection unit enacts the cacheable signal.

ECC:

Error Correcting codes, e.g., SEC-DED, are widely used to secure information against SEUs. Be that as it may, the nearness of SEMUs makes these codes inefficient to be utilized as a part of

exceedingly solid frameworks . Utilizing all the more capable error correcting code, e.g., double error correction triple error detection(DEC-TED),single-error correctiondouble-error detection double-adjacent-error correction(SEC-DED-DAEC), and interleaving ECCs, can successfully be utilized to secure the framework against SEMUs . Interleaving ECCs, in any case, forces serious energy,area and/ or performance execution overheads and the framework still stays helpless against multiple errors more prominent than two bit-flops for DEC-TED and SEC-DED-DAEC codes.

Difficulties in Design and Implementation of CADS

In spite of high level of reliability given by the CADS, a few difficulties in outline and execution of the proposed architecture should be addressed.

1)How to ensure the full duplication of dirty SPM lines.

2)How to reestablish error free replicas of dirty data from cache.

3)How to recognize perfect and dirty lines in SPM.

- Guaranteeing Full Duplication of Dirty SPM Lines:

One major disadvantage to replicate the SPM data lines in the cache is the likelihood of these replicas from the cache upon a cache miss. Once a replica line is removed from the cache, SPM ends up helpless against soft errors as the first dirty data line in SPM has no replica any more.

As a result of dynamic exchanging of data lines between the SPM and the main memory, a dirty line in the SPM might be replaced by another line. For write operations into the SPM, just a single out of the accompanying three cases requires the locking task.

- A Write Operation Into an Unreplicated Clean Word:

For this situation, the clean word ends up dirty and a cache line is assigned for repeating the word. The apportioned cache line should be locked.

- A Write Operation Into a Dirty Word:

For this situation, a cache line has just been allocated to the dirty wordwhat's more, it is sufficient to refresh the data word in the SPM and its replica in the cache. The cache line has just been locked and no further locking activity is required.

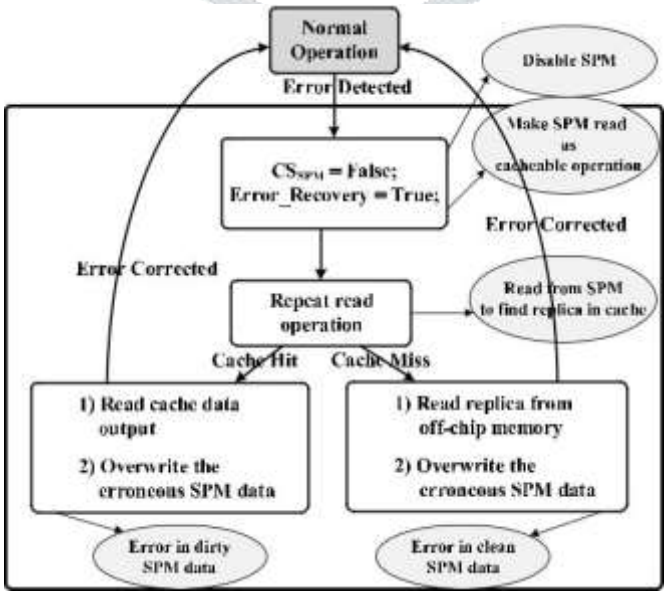

Fig. 3. Error correction procedure in CADS.

- A Write Operation Into an Already Replicated Word:

A reserve line assigned for replication contains the replica of eight neighboring SPM words. A cache line has just been designated and locked for a clean SPM word that no less than one of its eight nearby words (8– 1) isdirty. Since the SPM space is for the most part apportioned to clusters got to in loops, the majority for the greater part of clean SPM words thathave just been allotted and kept (seven out of eight in the best case).

- Restoring Error-Free Replica of Dirty Data Lines From Cache:

Upon mistake identification in a dirty line, the processor should read the error free copy of the dirty line from the cache and overwrite the incorrect line of the SPM by the error free replica. Since read access from the SPM are not cacheable tasks, regular read access can't reestablish the reinforcement duplicate accessible in the cache. To read a error free line from the cacheduring error recovery, after enactment of error detection from the SPM, the SPM is debilitated by inactivating SPM chip select and error recovery signal is initiated in the following clock cycle. Initiation of the error recovery signal makes the read task from the SPM address space as a cacheable activity keeping in mind the end goal to constrain the cache to convey the copy data line to the processor.

- Distinguishing Between Clean and Dirty Data Lines in SPM:

A conventional method to recognize the clean and dirty information is to utilize a dirty flag bit for each SPM line.In this architecture, all dirty SPM lines are put away in the cache and henceforth the cache contains no clean SPM line. At the point when a error is distinguished, we first look into the cache to discover the replica. In the event that a cache hit happens, the incorrect information are a dirty line and will be revised utilizing its copy in the cache. Something else, a cache miss demonstrates that the incorrect line is perfect and the copy will be read from the main memory.

The error correction routine is initiated subsequent to identifying a Clean error on a read a operation from the SPM. The activation of error detection signal is an hardware interrupt to the processor that requires quick consideration. To respond to this interrupt , the processor suspends its current exercises, saves its state, and executes an interrupt handler working as indicated by error correction procedure.

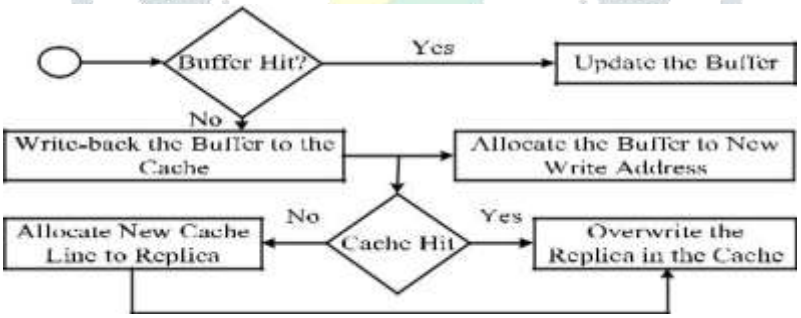

Fig. 4. Control flow of a write access to SPM in BCADS

Design Optimization

To reproduce all dirty SPM lines in the cache, each write access to the SPM needs an extra access to the store. Additional cache access are forced in light of refreshing the replicas in the cache or dispensing new cache lines for the allocation of SPM lines on the off chance that they are not as of now accessible.

For every write operation to the SPM, the write address is compared with the address of data line stored in the buffer. If the replica is already in the buffer, which can be interpreted as a buffer hit, the buffer entry will be updated. Otherwise, on a buffer miss, after writing back the buffer entry to the cache, the buffer will be allocated to the new data written to the SPM. For the write-back data line from the buffer to the cache, if the replica line is already allocated in the cache, the line will be rewritten. Otherwise, a new cache line is selected for replication before writing back the buffer entry.

BCADS can significantly reduce the extra cache accesses by utilizing the locality of references in SPM access. The SPM lines are mainly allocated to data arrays accessed in loops, which have a highly localized access pattern.

DMA:

A direct memory access (DMA) is an operation in which data is copied (transported) from one resource to another resource in a computer system without the involvement of the CPU. The task of a DMA-controller (DMAC) is to execute the copy operation of data from one resource location to another. The copy of data can be performed from: - I/O-device to memory - memory to I/O-device - memory to memory - I/O-device to I/O-device A DMAC is an independent (from CPU) resource of a computer system added for the concurrent execution of DMA-operations. The first two operation modes are 'read from' and 'write to' transfers of an I/O-device to the main memory, which are the common operation of a DMA-controller. The other two operations are slightly more difficult to implement and most DMA-controllers do not implement device to device transfers. The DMAC replaces the CPU for the transfer task of data from the I/O-device to the main memory (or vice versa) which otherwise would have been executed by the CPU using the programmed input output (PIO) mode. PIO is realized by a small instruction sequence executed by the processor to copy data. The 'memcpy' function supplied by the system is such a PIO operation. The DMAC is a master/slave resource on the system bus, because it must supply the addresses for the resources being involved in a DMA transfer. It requests the bus whenever a data value is available for transport, which is signaled from the device by the REQ signal. The functional unit DMAC may be integrated into other functional units in a computer system, e.g. the memory controller, the south bridge, or directly into an I/O-device. simplified logical structure of a system with DMA Arbiter CPU Memory DMA controller I/O device.

Simulation And Synthesis Results:

The simulation results of cache memory, cache controller,scratchpad memory,DMA can be shown in below figures.

Cache Memory:

Number of Slice registers: 8

Number of Slice LUTs: 10

Number of fully LUT_FF pairs: 8

Number of Bonded IOBs: 4283

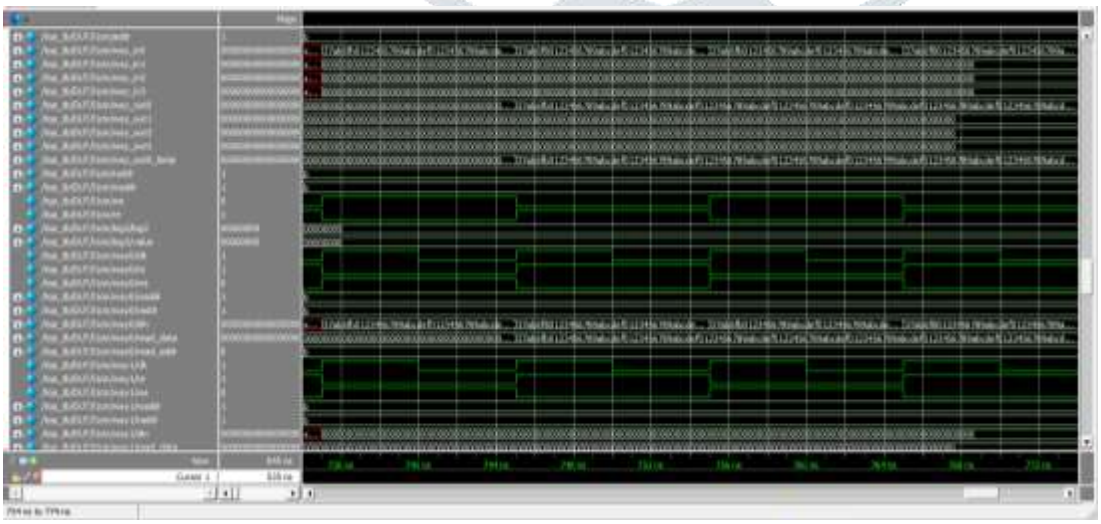

Fig5 Synthesis of Cache Memory

The input pins are:

clk,rst,enable,wr_rdb,addr,way_in0,way_in1,way_in2,way_in3,raddr,waddr

The output pins are:

way_out0,way_out1,way_out2,way_out3.

Cache controller:

Number of Slice Registers: 2160

Number of Slice LUTs: 9318

Number of fully used LUT-FF pairs: 2158

Number of Bonded IOBs: 5375

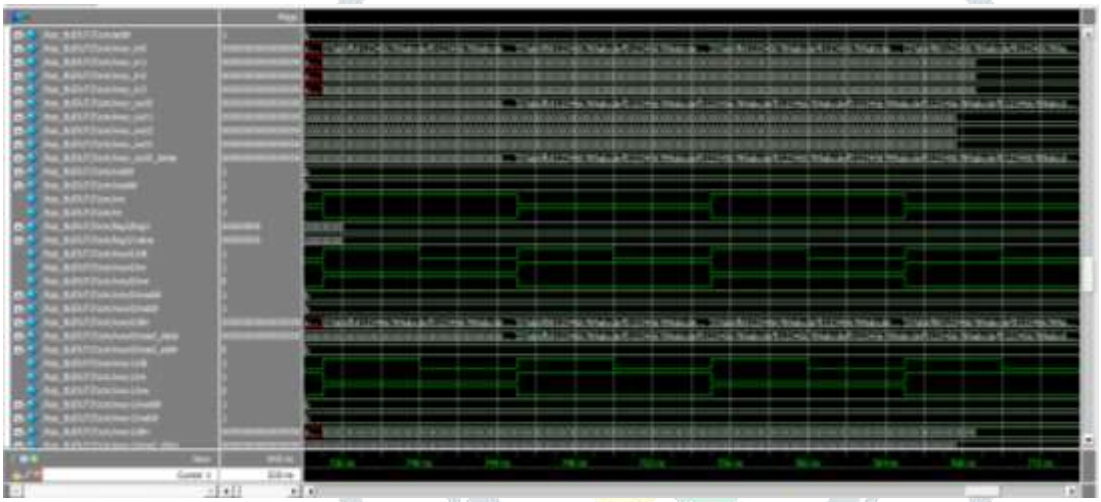

Fig 6 Synthesis of Cache Controller

The input pins are:

clk,rst,wr_rdb_from_higher_level,en_from_higher_level,wdata_from_higher_level,addr_from_higher_level,rdata_from_lower_level,rdata_valid_from_lower_level

rdata_to_higher_level,request_done_to_higher_level,hit,miss,cache_rdata_valid.

The output pins are:
wr_rdb_to_lower_level,en_to_lower_level,wdata_to_lower_level,addr_to_lower_level,,cache_wr_rdb,cache_en,cache_addr,cache_wdata0,cache_wdata1,cache_wdata2,cache_wdata3,cache_rdata0,cache_rdata1,cache_rdata2,cache_rdata3.

These are the parameters
cache_size,cache_line_size,num_ways,num_cache_lines,num_sets,tag_bits,
index_bits,lru_bits,offset_bits, memory_line_size.

In idle state,no read or write operation is done, else read or write operation is done.If en_from_higher_level is done,then read operation is accessed.

The address from higher level should match with the request from the processor.

If it is matched,it is a hit,or else miss.

For read operation, the address from the request should match with the memory address.For write operation, the address request should match with the cache address, so that the address is copied.

Scratchpad Memory:

Number of Slice registers: 23
Number of Slice LUTs: 109
Number of fully LUT_FF pairs: 18
Number of Bonded IOBs: 143

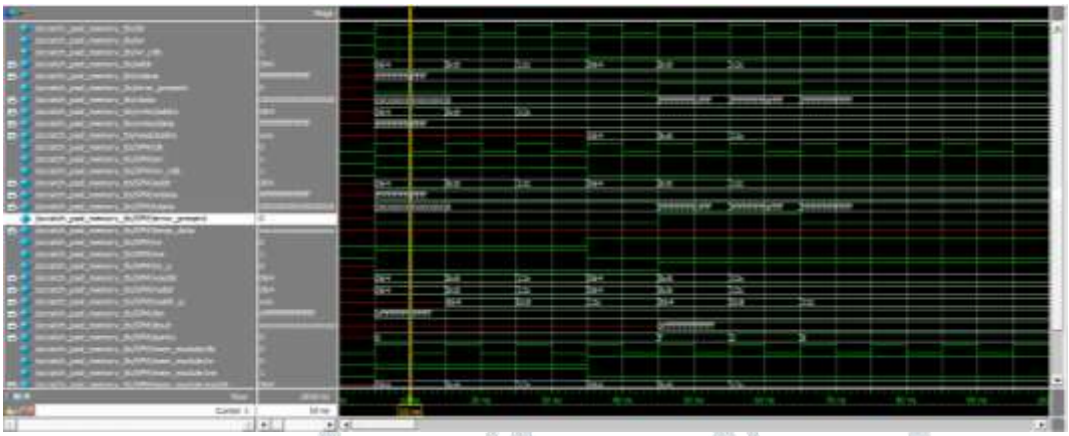

Fig7 Synthesis of Scratchpad memory

The input pins are clk,en,wr_rdb,addr,wdata. The output pins are rdata,error_present.We use parameters like address bits,data width,parity bits,memory depth.

DMA:

In this DMA simulation,we use

Number of Slice registers: 231

Number of Slice LUTs: 258

Number of fully LUT_FF pairs: 129

Number of Bonded IOBs: 217

Fig 12 Synthesis of DMA

The input pins are:

clk,rst,program_dma,reg_addr,reg_data,mem_read,mem_write,mem_addr,mem_wdata,mem_rdata,mem_rdata_valid,

The output pins are:

dma_done,spm_write,spm_addr,spm_read,spm_wdata,spm_rdata.

For example, source address is 32 bit data and destintion address is 32 bit data,transfer length is 8 bits.

If source address and destination address completely matched,then read or write operation is done.If both addresses matches,dma_done signal is enabled.

The SPM address is calculated as addition of source/destination address and transfer count.

Advantages:

The Cache assisted duplicated SPM is the best method to resolve soft errors in SPMs with very little area over head and performance overheads and no change in SPM access policies.

Applications:

In embedded processors for use in applications in IOT and in DSP processors where processing happens on large arrays of data.

Future scope:

Implement a Robust ECC scheme using less number of redundant bits to detect more number of errors.

REFERENCES

[1] (2015). The ARM Corporation. [Online]. Available: http://www.arm.com/products/processors/cortex-r

[2] (2015). The Renesas Electronics Corporation. [Online]. Avail-able: http://www.renesas.com/products/mpumcu/superh/sh7780/sh7785/ index.jsp

[3] W. Zhang, S. Gurumurthi, M. Kandemir, and A. Sivasubramaniam, "ICR: In-cache replication for enhancing data cache reliability," in Proc. Int. Conf. Dependable Syst. Netw. (DSN), Jun. 2003, pp. 291–300.

[4] M. Manoochehri, M. Annavaram, and M. Dubois, "CPPC: Correctable parity protected cache," in Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA), Jun. 2011, pp. 223–234.

[5] L. A. D. Bathen and N. D. Dutt, "Embedded RAIDs-on-chip for bus-based chip-multiprocessors," ACM Trans. Embedded Comput. Syst., vol. 13, no. 4, pp. 83:1–83:36, 2014.

[6] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in Proc. IEEE Int. Rel. Phys. Symp. (IRPS), Apr. 2011,

pp. 5B.4.1–5B.4.7.

[7] N. N. Sadler and D. J. Sorin, "Choosing an error protection scheme for a microprocessor's L1 data cache," in Proc. Int. Conf. Comput. Design (ICCD), Oct. 2007, pp. 499–505.

[8] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. Hoe, "Multi-bit error tolerant caches using two-dimensional error coding," in Proc. 40th Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO), Dec. 2007,

pp. 197–209.

[9] H. Farbeh and S. G. Miremadi, "PSP-cache: A low-cost fault-tolerant cache memory architecture," in Proc. Design, Autom. Test Eur. (DATE), Mar. 2014, pp. 1–4.

[10] A. Neale and M. Sachdev, "A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory," IEEE Trans. Device Mater. Rel., vol. 13, no. 1, pp. 223–230, Mar. 2013.

Authors profile:

ASAPU HARIKA Received her bachelor's degree in 2015 in Electronics and Communication Engineering from Stanley College of Engineering and Technology for Women, Abids,Hyderabad, India, which is affiliated to Osmania University, India. Her areas of interest include VLSI design. She is pursuing her M-Tech in VLSI SYSTEM DESIGN from CMR Institute of Technology.

Mr.A Sai Kumar Goud is working as associate professor in ECE departmentof CMRIT. He completed his B.Tech. from JNTUH Hyderabad and M.E. from University College of Engineering, Osmania University. He is currently in the submission phase of his Ph.D. thesis in College of Engineering, Osmania University, Hyderabad. He has an academic experience of 16 years In Teaching as well as research. He worked in various reputed engineering colleges as assistant professor, associate professor and HOD of ECE. He guided more than 100 B.Tech. Students and more than 20 M.Tech. students in their major projects. He has qualified UGC-NET. He is ratified as associate professor by JNTUH. His areas of research and interest are wireless communications & signal processing and VLSI. He is a life member of ISTE and published several national and international journals.

Mr. S Pradeep Kumar Reddy B.E, M.Tech (Ph.D), Working as Associate Professor in CMR institute of technology.He has 15 Years of Experience in teaching field. His area of interest is Image processing.