An Alternative Organization of Defect Map for Defect-Resilient Embedded On-Chip Memories

Kang Yi[1], Shih-Yang Cheng[2], Young-Hwan Park[2], Fadi Kurdahi[2], and Ahmed Eltawil[2]

[1] School of Computer Sceince and Electronic Engineering, Handong Global University, Pohang, Korea
yk@handong.edu
[2] Department of EECS, University of California, Irvine, CA 92697-265
{shihyanc,younghwp,kurdahi,aeltawil}@uci.edu

Abstract. In this paper, we propose the low power and low area defect map organization for the defect-resilient embedded memory system for multimedia SOCs. Existing approach to build defect map of embedded memories is based on the CAM (Content Addressable Memory) organization. But, it consumes too much power and relatively large chip area. It may be serious problem in the near future for very deep submicron technologies. Therefore, we propose the SRAM-based defect map organization to reduce both the power consumption and chip area. We also develop new defect map access algorithm to minimize the number of defect map access operations to save power. Our estimation results show the new scheme based on SRAM defect map organization consumes only 1/4 times of power at BER=1.0% compared with the power overhead by the existing approach.

Keywords: Embedded memory Yield, Defect Map, Memory Error Resilient Design, Video error concealment.

1 Introduction

The memory hungry application is becoming the dominant portion of the SOC market because of highly increasing demands on multimedia applications. According to the 2001 International Technology Roadmap for Semiconductor, the embedded memories are going to occupy from 54% to 94% of silicon real estate by year 2014 [1,2] as shown in Figure 1. In addition, the ever-shrinking geometry of semiconductor devices is pushing the memory parts into a single chip integrated with core logic parts because of many practical benefits such as manufacturing cost reduction, performance enhancements, and much more power saving.

However, it is well known problem that the large embedded on-chip memory with very deep submicron technology will be suffering from high defect density resulting in low SOC chip yield and high production cost.

Even though there are several existing approaches for this memory recovering from defects problem, these approaches with redundancies require too much area overhead for the memory defect density in the near future. In [3] we can see the

L. Choi, Y. Paek, and S. Cho (Eds.): ACSAC 2007, LNCS 4697, pp. 102–113, 2007.
© Springer-Verlag Berlin Heidelberg 2007

prediction of the defect densities that are 1-2 orders or magnitude higher than today's defect density. In [4] the existing approach with redundancy scheme to address 0.1% defect density shows a huge cost of 70% area overhead.

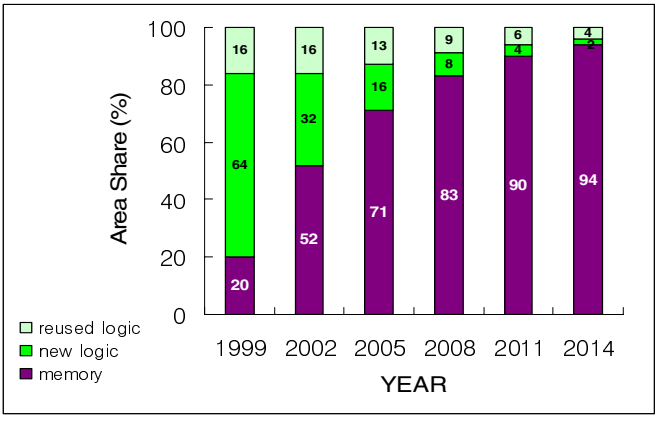


Fig. 1. Area of memory portion in SoC design trend

An innovative approach to this embedded memory data recovering problem with highly defect density for the near future technology was proposed by [5] and [6] focusing on the multimedia application. The new approach observes that multimedia applications have the information redundancy in themselves like spatial and/or temporal locality. The new approach is based on the defect map which records all the defect memory cell location and post processing to recover corrupted data from defective data location. But, the overhead of the new approach is defect map power consumption.

Therefore, in this paper we propose an alternative defect map organization that has the same functionality as previously proposed system while it saves much of the power consumption. The key idea is using of SRAM instead of CAM (Content Addressable Memory) and reduction defect map read operation frequency drastically with new memory lookup strategy.

2 Summary of the Previous Work

2.1 Overview of Filtering Scheme with Defect Map

By utilizing the redundant information in multimedia application data themselves, [6] developed an image data recovery method from corrupted memory targeting at H.264 decoder system. According to the previous work from [6] they achieved high visual quality as well as high PSNR values even with simple image filters if the defect pixel location is known by defect map. The works in [6] recovers corrupted moving pictures with memory defects up to 1.0% bit defect density which is quite higher density than necessary even in the foreseeable future technology. Figure 2 shows the system block diagram of the approach.

The defect map in this system is constructed by built-in-self-testing process at the time of system power up booting phase. Once the defect map is constructed, the defect map lookup is requested for every read operation for the pixel data stored in a frame buffers to see if the location is defect or not. Whenever an H.264 decoder tries to read the defect location, image filter is applied to get the properly estimated data for the pixel. In this filtering process, neighboring pixel data in the frame buffer (DPB memory) may be required, too. Figure 3 shows the visual quality of corrupted and recovered image by the filtering with defect map approach. The recovery result is almost very good.

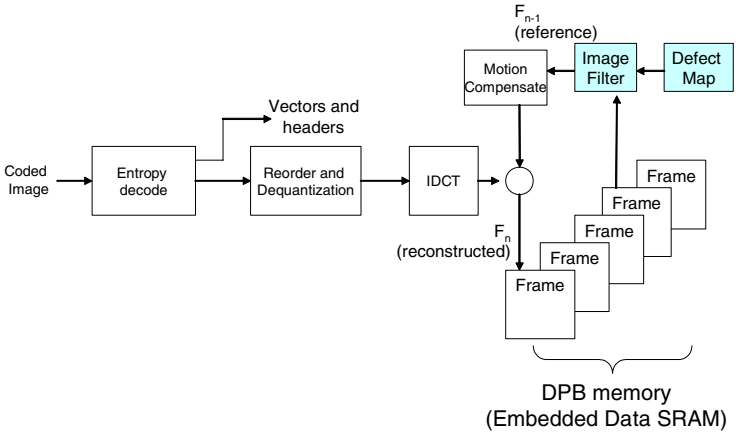
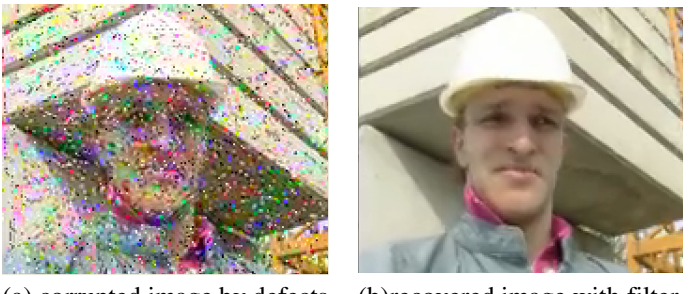


Fig. 2. The overall architecture for defect-resilient multimedia data memory with H.264 decoder system



(a) corrupted image by defects (b)recovered image with filter

Fig. 3. Comparison of Images before and after recovery by [6] at BER=1.0%

2.2 The Problem with Filtering Scheme with CAM-Based Defectmap

Figure 4 shows the CAM defect map organization. It emphasize that defect map reference is required for every pixel value reading. The problem with the defect map is that at higher defect rates the defect map power consumption increases and the defect map area overhead is no more negligible. Figure 5 shows this problem with

power overhead of filtering scheme for different defect densities in memory bits. The problem stems from the fact defect map size is proportional to the pixel error rates. The relationship between pixel error rates (PER) and bit error rates (BER) are shown equation (1).

$$PER = 1 - (1 - BER)^{PIXEL\_DEPTH} \quad (1)$$

The PIXEL_DEPTH in equation (1) is the number of bits for each pixel which is usually 8. The power consumption of the defect map is proportional to the size of defect map because of CAM nature.

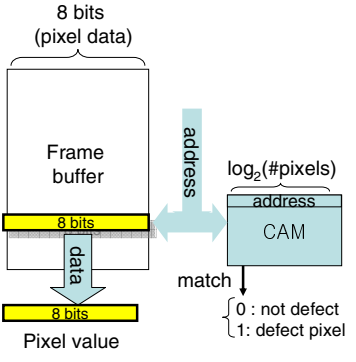


Fig. 4. CAM-based defect map organization

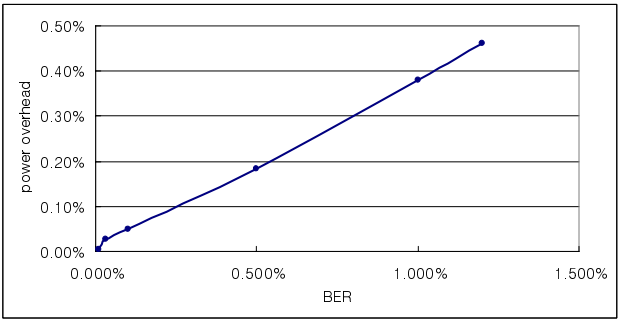


Fig. 5. Power overhead by filtering scheme with CAM-based defect map organization

This experimental power consumption data is based on the numbers in [7] and assumes the implementation with 90 nm technology and VGA sized image with 5 reference frames. In the Figure 6 we analyzed the source of power consumption. We observe that defect map power consumption accounts for more than 80% of the total filtering scheme power consumption at higher defect densities. Note that the BER=0.01% is high enough to meet the current technology requirements but, it is expected the BER may be higher than 0.1% under the coming very deep submicron technology. Therefore, in order to reduce the power overhead of filtering scheme for

the foreseeable future technology we have to find a way to reduce the defect map power consumption. In this paper, we propose the use of SRAM for the defect map construction rather than CAM-based defect map organization.

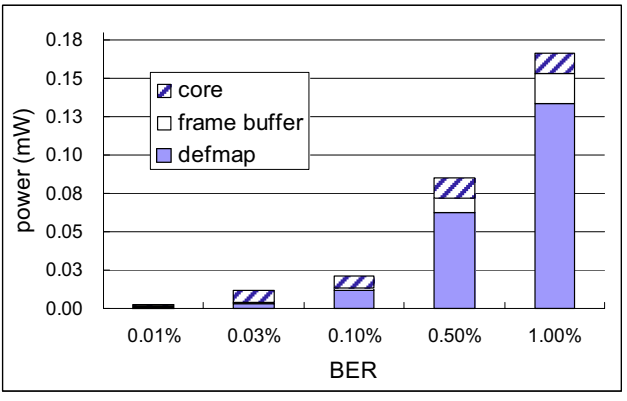


Fig. 6. Power consumption share by filtering scheme with CAM- based defect map organization

3 Alternative Defect Maps

3.1 The Idea of SRAM-Based Defect Map Organization

We compare the old defect map with our new SRAM-based one in Figure 7. In this new defect map scheme, we use SRAM instead of CAM to save power. We store the addresses of defective pixels in the SRAM. And, we look up the defect map for every pixel data reading operation to find if there is any entry in the defect map that has the same given address. But, the problem with the SRAM-based approach is that we need to search all the SRAM memory to find a specific SRAM address having specific frame buffer address. What makes it worse is that SRAM only performs the search operation one by one manner while CAM performs a parallel search with contents. To avoid the exhaustive search in the defect map we assume the following two things.

(1) The addresses of defect pixels stored in the SRAM are all sorted in an ascending order.
(2) The frame buffer read operation for pixel value acquisition is performed by an address issued in an ascending order.

The above two assumptions can be realized by (1) well organizing the memory address ordering and (2) by address sorting at the time of defect map construction which occurs only one time per system power up. In this scheme we also use special registers called “defectmap_pointer” register and “defect_pixel_addr” register. The defectmap_pointer register points to the defect map location that contains the address of the frame buffer where the next defective pixel is located. Defect_pixel_addr represent the defect map content currently pointed by defectmap_pointer. Under the two assumptions above, we develop an SRAM-based defect map look up algorithm as shown in Figure 8.

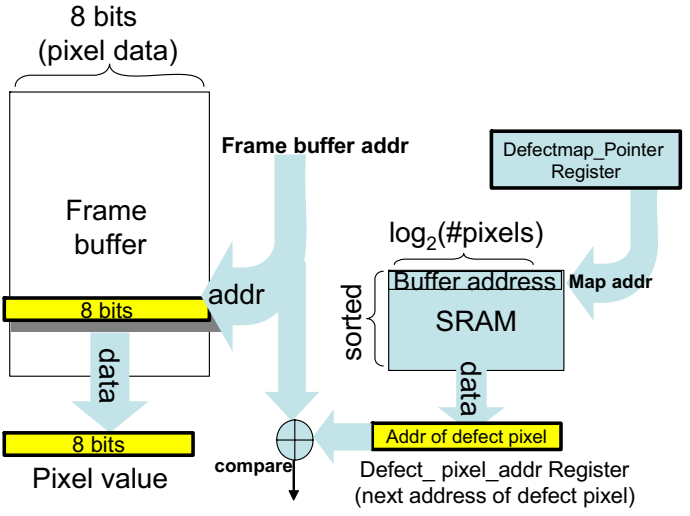


Fig. 7. Basic SRAM-based defect map organization

```
function DefMap_LookUp (frame_buffer_address : address)
  if(*defectmap_pointer == UNINITIALIZED) then
         defectmap_pointer = 0;
         defect_pixel_addr = *defectmap_pointer;
  end
  if (frame_buffer_address < defect_pixel_addr) then
         return DEFECT_NOT_FOUND;
  else if (frame_buffer_address == defect_pixel_addr) then
         defectmap_pointer = defectmap_pointer + 1 ;
         if (defectmap_pointer > MAX_ADDR) then
             defectmap_pointer = 0;
         end
         defect_pixel_addr = *defectmap_pointer ;
         return DEFECT_FOUND;
  end ;
  else begin
      return DEFECT_NOT_FOUND
  end
```

Fig. 8. SRAM-based Defect map lookup algorithm

The first line of the algorithm describes the defectmap pointer register and defect_pixel_addr initialization procedure. The following "if statement" determines whether the given frame buffer address is the defect location or not by comparing the given address with the defectmap output register. If both matches it means given address is defect location and increase the defect pointer and read out the content from the defect map which is the next defect pixel address in the frame buffer.

With this algorithm, we achieve the sequential read of defect map per frame. That means the core H.264 decoder reads defect map only as many time as the number of defective pixels in the frame buffer. Therefore, we can save the defectmap power consumption drastically with SRAM-based defect map organization due to (1) the less power consumption by SRAM circuit than CAM circuit per access and (2) the less number of defect map access by the new defect map access algorithm.

3.2 A Consideration for Image Filters: Enhanced SRAM-Based Organization

We still have the problem with the SRAM-based defect map organization proposed in previous subsection when we care about the image filters. Some filters developed in [6] require the surrounding pixel values neighboring the defect pixel. And, some of them also need to find whether each of neighboring pixels is in defect location or not. In order to provide this neighboring pixel defectiveness information we added more bits to each of the entries in the SRAM-based defect map. We show the modified organization in Figure 9.

The enhanced organization allocates more space of B bits slot per pixel value for the neighborhood pixel information of defectiveness. Each of bit in the extra space The B is from 0 to 8 where 0 means the image filter does not care about the defectiveness of neighboring pixels and 8 means the image filter needs every

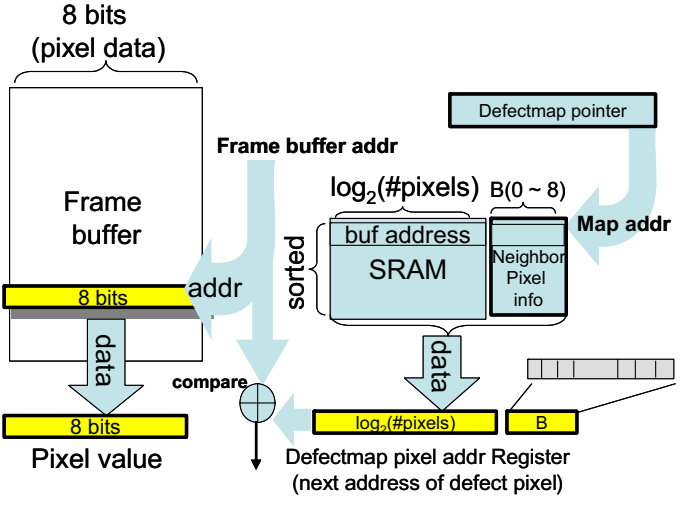


Fig. 9. Enhanced SRAM -based defect map organization

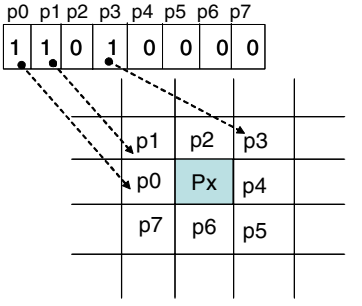


Fig. 10. Representation of defectiveness information of neighbor pixels around the defect pixel

neighboring pixel defectiveness information. Figure 10 demonstrates the meaning of each bit in the extra bits where B=8. If the left-most bit (1st bit), the 2nd, and the 4th bits in the extra field are '1's, the pixel p0, p1 and, p3 are defect pixels.

3.3 Simpler Defectmap Organization: Flag Bit Approach

Sometimes, simple strategy is the best strategy under a special condition. We propose another alternative defect map organization named "Flag" approach which can be effective when defect density is too high for separate defect map memory. Figure 11 shows the "Flag-based" defect map organization.

This approach adds one flag-bit for each pixel value component. Each of the flag-bit whose value is '1' means the corresponding entry of the frame buffer has one or more defect bits per entry of each frame buffer location. Its area overhead is 1/8 =12.5% regardless of defect density. But, it has the benefits over SRAM-based method : (1) very simple organization, (2) low power consumption comparable to enhanced SRAM-based approach, (3) low area overhead at very high defect density (BER >= 0.5%).

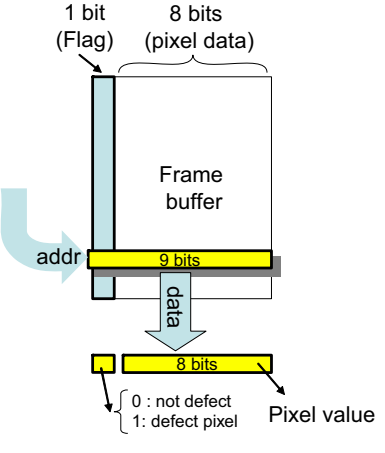


Fig. 11. Flag-based defect map organization

4 Quantitative Evaluation

4.1 Memory Area Overhead Comparison

The following equation (2) through (4) is for the defectmap area overhead for CAM-based defect map and equation (5) through (6) is for the enhanced SRAM-based defect map with B more extra bits per entry

$$Area_{FrameBuffer} = 8 \times Num\_Pixels \times Area_{SRAM\_CELL} \tag{2}$$

$$Area_{CAM} = \log_2(Num\_Pixels) \times Area_{CAM\_CELL} \times (Num\_Pixels \times PER). \tag{3}$$

$$Overhead_{CAM} = \frac{Area_{CAM}}{Area_{framebuffer}} = \log_2(Num\_Pixels)/8 \times (9/6) \times PER. \tag{4}$$

$$Area_{ESRAM\_MAP} = (\log_2(Num\_Pixels) + B) \times Area_{SRAM} \times (Num\_Pixels \times PER). \tag{5}$$

$$Overhead_{ESRAM\_MAP} = \frac{Area_{ESRAM\_MAP}}{Area_{Frambuffer}} = (\log_2(Num\_Pixels) + B)/8 \times PER. \tag{6}$$

In the equation PER is the value from equation (1). And we assume the ratio of CAM cell area over SRAM cell area 9/6 because CAM cell is composed of 9 TRs while SRAM is of 6 TRs. Based on the equations above we show the area overhead comparison graph in Figure 12 for CAM, Enhanced SRAM with B=0, Enhanced SRAM with B=8, and Flag approach. Below the high density defects (BER < 0.5%) SRAM-based defect map requires the least area overhead among different types. Flag-based defect map requires the least area overhead at higher defect densities.

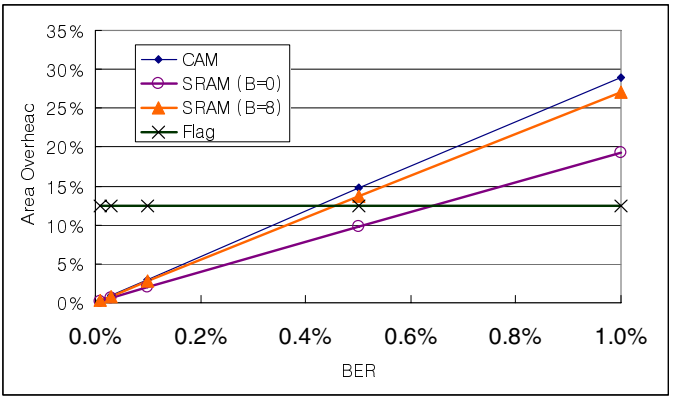


Fig. 12. Area overhead of different types of defect maps

4.2 Power Overhead Comparison

To estimate the power consumption with filtering scheme we scaled the published memory access frequency data from [8] to fit QCIF image with 30 fps and we used the H.264 decoder core power from [9] and estimated our filter core power

consumption by the power estimation tool. With the new defect map organization, we can expect quite low power consumption. Figure 13 shows the power consumption by filtering scheme with the enhanced SRAM-based defect map (B=8) for different BERs. Compared with the graph in Figure 6, we see the reduction of the relative power of defect map as well as the reduction of the whole power consumption reduction by filtering scheme. The defect map consumes less than 13% of the total power used by filtering scheme with SRAM-based approach while more than 80% of total filtering power is used by defect map access at BER=1.0%. Figure 14 shows the power consumption by the filtering scheme with the Flag-based defect map at different BERs. We see that the defect map power consumption is constant regardless of BERs while frame buffer consumes the power proportional to the defect rates. The overall power consumption of Flag-based defect map is slightly larger than that of SRAM-based defect map.

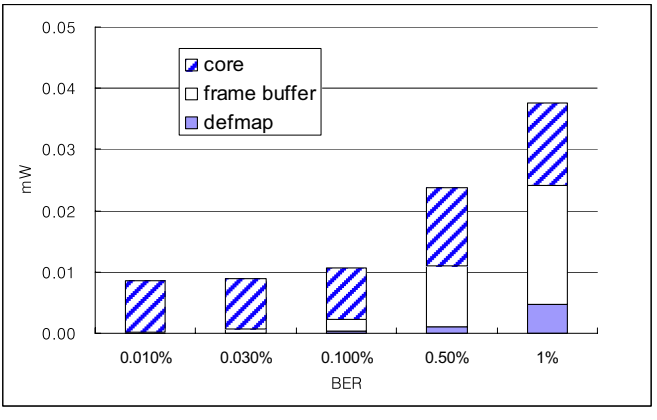


Fig. 13. The power consumption share by filtering scheme with SRAM-based defect map organization

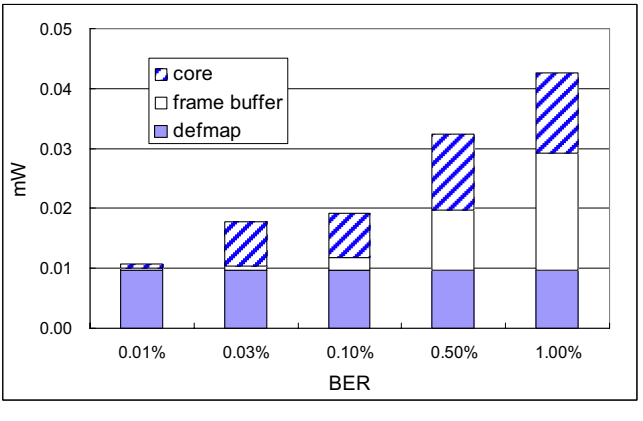


Fig. 14. The power consumption share by filtering scheme with Flag-based defect map organization

Now, in Figure 15 we compare the power consumption overhead by filtering scheme with different defect map organizations (CAM, SRAM, Flag) at different BERs. At BER=1.0% the power overhead by filtering with SRAM-based defect map and Flag-based defect map are less than 1/4 times of the power overhead by filtering with CAM-based defect map. Up to 0.1% BER the SRAM-based defect map is the best in terms of area and power overhead. At very high defect density beyond 0.5% BER and upto 1.0% BER, the SRAM-based defect map organization shows still slightly better performance than the Flag-based organization in terms of power overhead but, Flag-based approach is better than any other approaches considering the area overhead.

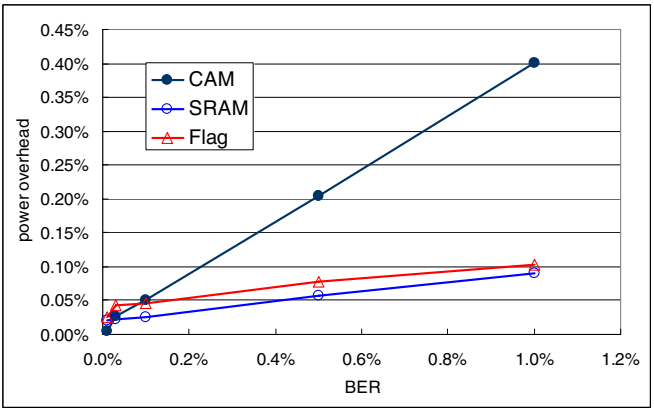


Fig. 15. Power overhead comparison by different defect map organizations at different BERs

5 Conclusion

We focus on the new memory defect recovery method for the multimedia application published in [6] that utilizes the defect map and simple image filters. We found that the defect map in [6] will suffer from the power consumption at higher defect density in the near future. In this paper we propose alternative defect map organizations that can save power consumption by 4 times less than the CAM-based defect map organization by using SRAM –based and Flag-based approach.

References

1. Shoukourian, S., Vardanian, V., Zorian, Y.: "SoC yield optimization via an embedded-memory test and repair infrastructure". IEEE Design & Test of Computers 21(3), 200–207 (2004)
2. http://public.itrs.net
3. Gupta, T., Jayatissa, A.H.: Recent advances in nanotechnology: key issues & potential problem areas. In: Proceedings of IEEE Conference on Nanotechnology, vol. 2, pp. 469–472 (2000)

4. Agarwal, A., Paul, B.C., Mahmoodi, H., Datta, A., Roy, K.: “A process-tolerant cache architecture for improved yield in nanoscale technologies”. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13(1), 27–38 (2005)
5. Kurdahi, F.J., Eltawil, A.M., Park, Y.-H., Kanj, R.N., Nassif, S.R.: “System-Level SRAM Yield Enhancement”. In: Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED 2006), pp. 179–184 (2006)
6. Yi, K., Jung, K.H., Cheng, S.-Y., Park, Y.-H., Kurdahi, F.J., Eltawil, A.: “Design and Analysis of Low Power Filter toward Defect-Resilient Embedded Memories for Multimedia SoC”. In: Proceedings of Asia-Pacific Computer Systems and Architecture Conference, pp. 300–313 (September 2006)
7. Pagiamtzis, K., Sheikholeslami, A.: “Contents-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey”. IEEE journal of Solid-State Circuits vol.41(3) (2006)
8. Horowitz, M., et al.: H.264/AVC Baseline Profile Decoder Complexity Analysis. In: IEEE Trans. on Circuits and Systems for Video Technology, vol. 13(7) (2003)
9. ARC, http://www.arc.com/subsystems/video/ARC_Video_pb.pdf