IRAM Design for Multimedia Applications

Bum-Sik Kim, Yun Ho Choi*, and Lee-Sup Kim

Korea Advance Institute of Science and Technology
E-mail:lskim@eekaist.kaist.ac.kr

*SAMSUNG Electronics Co., LTD.
E-mail:choi6003@samsung.co.kr

Abstract

There are strong demands for high speed and low power to realize systems on silicon. However, the current circuit design technologies for MPU and DRAM are based on their own optimized process technologies. It is necessary modify the circuit design technology for embedded memory logic after reviewing process technologies of MPU and DRAM. In this paper, we propose the modified design concept for IRAM for multimedia applications. After scrutinizing an example of multimedia applications, we propose a new IRAM architecture for multimedia applications. We are currently designing IRAM system which executes DCT of MP@ML in MPEG2 at 72.6MHz with 0.35μm CMOS EML (Embedded Memory and Logic) process with 4-metal layers. Also, we propose that the DRAM block should be divided as many blocks as possible to prevent DRAM block from being the bottleneck of the system performance. In addition, we propose that by using the new EML process technology, the DRAM circuits must be modified to use wide address buses and data buses.

1 Introduction

There have been many advances in the VLSI technology. The scaling of minimum design feature and the improvements of design technology have helped the speed-up of microprocessors as shown in Figure 1. Also, the scaling of minimum feature has enabled the new microprocessor architecture to get the high performance such as VLIW and SIMD. The improvements of fabrication technology enables the increase of DRAM density by Moore's law. Also, the design technique improves the speed of DRAM and generate new kinds of DRAM type for high performance applications as shown in Figure 2 [1].

However, the improvements of MPU(or DSP) performance require the large storage area of data with high speed in proportion to the size of software or data. Thus, there are some gaps between MPU and DRAM as listed below.

- Operation frequency. As shown in Figure 1 and 2, the clock frequency of MPU reaches at 600MHz and the clock frequency of DRAM is limited by 150MHz. MPU is 4-times faster than DRAM.

- Bus and Address width While MPU is the performance-driven semiconductor device, DRAM is the cost- driven semiconductor devices. Thus, the number of I/O data pins and address pins in DRAM is limited by the package type(related to costs). The cost also limits the width of address and data bus in DRAM chip.

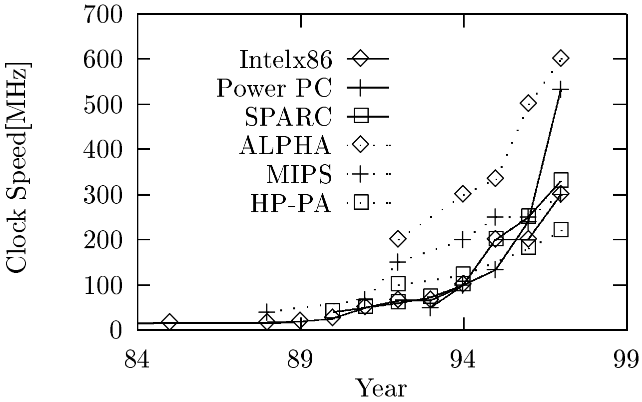


Figure 1: Maximum Clock Speed of MPU for each Series

To overcome the above gaps, there have been many improvements such as virtual memory system and cache memory system. Recently, the researches for new DRAM circuits, e.g. Rambus DRAM, Sync-Link and SGRAM and etc., to reduce the gap between MPU and DRAM bandwidth are being widely used. In addition to the above gaps, there are some constraints for effective usage of MPU and DRAM for some applications.

- Granularity
  The size of DRAM for general purpose is fixed, e.g. 16Mb, 64Mb, 256Mb and etc. However, for special applications, the specific size of memory is necessary. In that case, as shown in Figure 3, the remaining memory(shaded area) is wasteful. For digital video applications, the required size of memory for one frame of one color is 1B×728×480≈0.35MB, i.e. 2.8Mb. If a conventional 4Mb DRAM is used, 1.2Mb is wasteful.

- Costs
  The aspect of cost has relations to the granularity and architecture. For the some applications, the size of general purpose DRAM is not optimum. We can find an example in consumer products such as a television. For

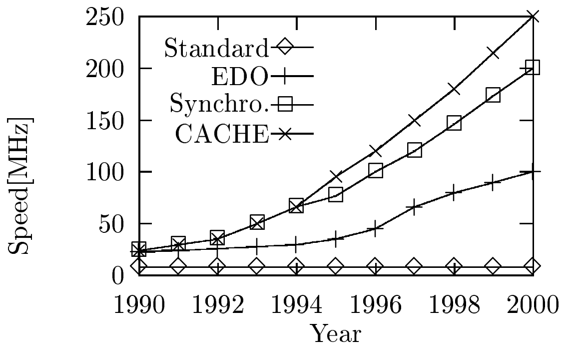


Figure 2: Comparison of Speed Capability of DRAMs

example, if a frame buffer requires 4.2M bits of memory and only 4M and 16M RAMs are available, then a 4.2M application specific memory could cost less than two 4M or one 16M DRAM as long as it is produced in large volume to cover the development and production costs. A hypothetical cost comparison of application specific DRAM and standard DRAM is shown in Table 1.

Table 1: Hypothetical Cost Comparison of Application Specific DRAM and Standard DRAM

Time Points	A year	A+1 year	A+2 year
16M DRAM cost	$20	$15	$10
4M DRAM cost	$5	$4	$2
4.2M DRAM cost	$8	$7	$6

- Power Dissipation
  The power dissipation in I/O pad of DRAM can be calculated using the equation (1). An example for 32-bit I/O bus width is shown in Figure 4.

$$P_{D,I/O} = N \times C_L V_{CC}^2 f \qquad (1)$$

  The power dissipation increases in proportion to the bandwidth and supply voltage of DRAM. To sustain high speed, the supply voltage must be kept as high as possible. There exists an contradiction between speed and dynamic power dissipation.

2 Motivation

Generally, SRAM has been used for the embedded memory system because of the convenience [2]. For example, high speed SRAM has been used as cache memory because of the performance in the general purpose microprocessors. However, the density of SRAM is much lower than DRAM. Thus, the researches on the integration of MPU and DRAM in a single chip becomes active. The researches that merge the DRAM and MPU(or DSP) in a single chip are focused on the high performance system such as MPPs and vector processors by architecture level approach [3] [4]. The architecture in [3] is focused on getting the high performance using vector processors and the architecture in [4] is similar to the multiprocessor board with wide bus between processor and main memory. The majority of the previous works are simple integration of MPU and DRAM based on MPU process or DRAM process [5]. However, the fabrication technology of MPU is different from that of DRAM as shown in Table 2. The target of MPU has been the performance, but, the target of DRAM has been density. Thus, the advances in design technology of MPU are mainly based on the multiple metal layer fabrication technology. However, the advances in technology of DRAM are based on the shape of DRAM cell and multiple polysilicon(silicide) layers. Since DRAM uses polysilicon(silicide) layers, the resistance of polysilicon causes increases of RC time delay.

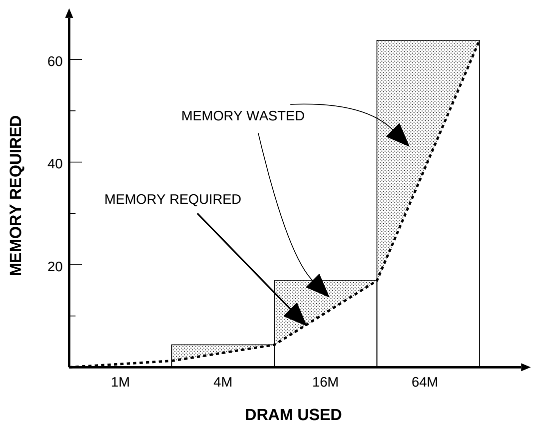


Figure 3: Silicon Wasted In Systems using a Non-Standard Memory

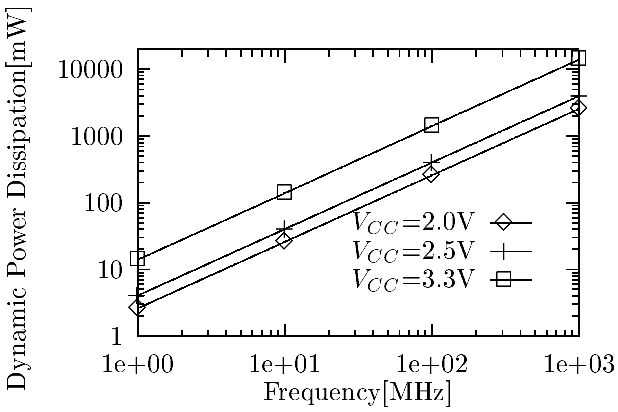


Figure 4: Power Dissipation in the Output Pin

Table 2: Process technologies of DRAM and MPU

Paper	Process
233MHz MPU [6]	5 metal
600MHz MPU [7]	6-metal
300MHz MPU [8]	4-metal
1G DRAM [9]	3-poly, $2TiS_2$, 2W, 2-Al
256Mb SDRAM [10]	3-poly, 2-W, 1Al-Cu
1Gb DRAM [11]	3-poly, 2-W, 2-Al

As listed in Table 2, there is a large gap between MPU and DRAM processes. The design concepts are quite different. Therefore, the simple integration of MPU and DRAM in an MPU fabrication process or a DRAM results in the performance degradation of one side. There are some activities to develop new process technology for embedded memory logic. But unless the architecture and circuits of MPU(or general purpose DSP) and DRAM are changed, no more improvements other than widening of bus width and increasing of data I/O frequency of DRAM are possible.
Since the architectures and circuits of MPU are optimally configured for the performance and those of DRAM for the density and the costs, radical changes in DRAM and MPU(or DSP) architectures and circuits are deemed to be inappropriate. Therefore, we adopted an approach, that is, marginal variations of DRAM and DSP architectures and circuits. In the digital video signal processing or 3-D graphic applications, the processor should deal with the large amount of data and computations. The data flows of the digital video signal processing and the 3-D graphics are deterministic. Thus, it is informative to check the key limitations of the conventional architectures of DRAM and MPU. Two serious problems in the conventional DRAM and MPU are listed below.

- Access time of DRAM is fixed except the I/O time
- MPU(or DSP) consumes larger power than DRAM except the I/O power for DRAM

In order to find the usefulness of the embedded memory system, we configured the optimized IRAM system for multimedia applications with new EML process technology available based on the ideas which are listed below.

- Wide bus can be connected to DRAM directly to have high bandwidth.
- Many VLSI components can be integrated on the same chip by using complicated process technology.
- Other circuit components(e.g. SRAM, ROM) can be modified to get the optimized high performance .
- Resistance of word line in DRAM circuits can be lowered by exploiting some additional metal lines available in a new EML fabrication technology.
- Row and column addresses of DRAM can be applied in the same time using wide address bus.
- More dense logic circuits can be designed using multiple polysilicon or silicide [12].

The necessary circuits to perform multimedia operations, such as barrel shifter and coefficient memory(as a ROM), in IRAM system are added as well.

3 Design of IRAM for Multimedia application

For the efficient usage of each block, the data processings of one multimedia application example are analyzed. And this example is designed to be the embedded memory system using the CMOS process shown in Table 3. In our design, the data processings are incorporated inside the chip as much as possible to maximally utilize the IRAM concept.

Table 3: CMOS EML Process for Embedded Memory System

Process	Logic	DRAM
L_{min}	0.35μm	0.35μm
Well	N-well	Triple well
Metal	4-layer	4-layer
Poly	1-poly	3-poly
V_{TN},V_{TP}	0.55V, -0.55V	0.55V, 0.45V, -0.55V
T_{ox}	8nm	8nm

3.1 Application Example of IRAM

As a test vehicle to verify the performance improvement in a merged DRAM logic, we designed a new architecture of IRAM for multimedia applications. DCT(Discrete Cosine Transform) is the most widely used core algorithm in image data compression standards such as MPEG. Therefore, In this paper, DCT algorithm is analyzed thoroughly in order to find the efficient architecture in video signal processing. In the high performance multimedia applications, if IRAM executes the DCT operation, main processor can execute at the same time the VLC(Variable Length Coding) operation and other control operation of the whole video compression / decompression system. The operation of DCT in MPEG can be divided into two sub-operations; one, the input of video signal input and subtraction of two frames in memory, the other, 8×8 block DCT operation. The subtraction operation of image frame is not DCT operation in a strict sense. The equations for 8-point DCT operation are shown as (1) and (2) [13].

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{3}$$

Equations (1) and (2) are converted to the computational sequence in Figure 5.
The two operations, an arithmetic operation between image frame and 8×8 2D-DCT operation, are visualized in Figure 6.
In Figure 6, the subtraction operation which is dependent on the type of a frame is optional. As shown in Figure 6, the processor needs at least two ALU units which can be operated independently. When the video signal data are inputted into the IRAM, the image frame for DCT operation are stored into the certain DRAM after the arithmetic operation between image frames. Therefore, it is needed to be equipped with at least two ALU units to deal with the DCT operations . Also, we must divide the memory block into four memory sub-blocks and each block is corresponding to one image frame. From the analysis of DCT operation , the operations which are needed in the IRAM are summarized as below.

$$\text{LOAD PE[0][0]} \sim \text{PE[7][7] to R[0]} \sim \text{R[63]}$$
$$\tfrac{1}{2}\,(\text{R[0]}\pm\text{R[7]}) \quad \text{to} \quad \text{R[64],R[65]}$$
$$\tfrac{1}{2}\,(\text{R[1]}\pm\text{R[6]}) \quad \text{to} \quad \text{R[66],R[67]}$$
$$\tfrac{1}{2}\,(\text{R[2]}\pm\text{R[5]}) \quad \text{to} \quad \text{R[68],R[69]}$$
$$\tfrac{1}{2}\,(\text{R[3]}\pm\text{R[4]}) \quad \text{to} \quad \text{R[70],R[71]}$$
$$y_0=\sum_{i=64,66,68,70} \tfrac{1}{\sqrt{2}}\text{R[i]}$$
$$y_1=c_1\text{R[65]}+c_3\text{R[67]}+c_5\text{R[69]}+c_7\text{R[71]}$$
$$y_2=c_2\text{R[64]}+c_6\text{R[66]}-c_6\text{R[68]}-c_2\text{R[70]}$$
$$y_3=c_3\text{R[65]}-c_7\text{R[67]}-c_1\text{R[69]}-c_5\text{R[71]}$$
$$y_4=c_4(\text{R[64]}-\text{R[66]}-\text{R[68]}+\text{R[70]})$$
$$y_5=c_5\text{R[65]}-c_1\text{R[67]}+c_7\text{R[69]}+c_3\text{R[71]}$$
$$y_6=c_6\text{R[64]}-c_2\text{R[66]}+c_2\text{R[68]}-c_6\text{R[70]}$$
$$y_7=c_7\text{R[65]}-c_5\text{R[67]}+c_3\text{R[69]}-c_1\text{R[71]}$$

Figure 5: Computing Sequences for 8-point DCT

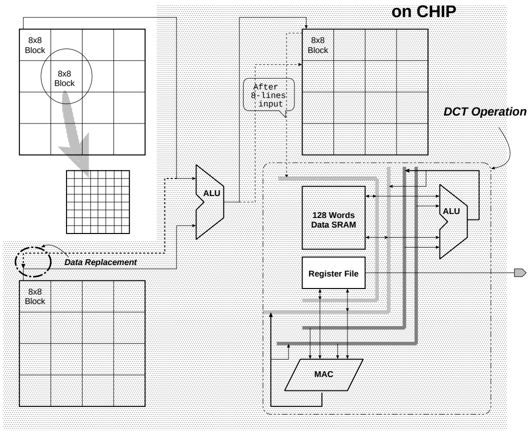


Figure 6: Data Flow in DCT of MPEG

- Arithmetic operation between the data from I/O bus and the data from the stored image in the DRAM
- 8×8 block 2D-DCT operation which is independent of I/O operation
- Frame store operation
- Output of the results of 2D-DCT operation

3.2 Operation of DRAM Block

As shown in Figure 6, DRAM block must be divided into more than three blocks not to be the bottleneck in the operation IRAM system. An operation example of DRAM block is shown in Figure 7. The mode in the example is a DCT operation in MPEG. The basic operation scheme of DRAM block is ping-pong operation. Control and status register for each block is necessary to monitor and control the each DRAM block independently. The details of DRAM Block will be discussed in the section 3.4.2.

3.3 Instruction Set Architecture

From the previous section, we can extract five types of instructions to deal with the DCT operation efficiently. The

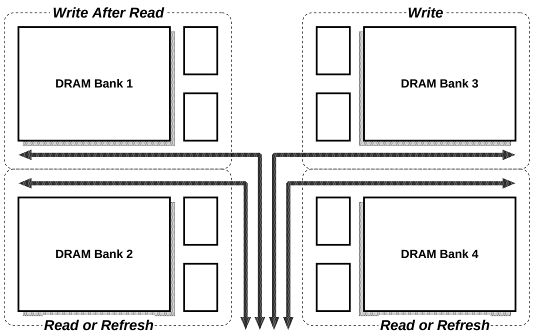


Figure 7: DRAM Operation Mode Example in DCT

key operations which the instruction set of IRAM system should cover are listed below.

- Execute ALU operation between the external data and the internal data and move the results into the internal register(or memory).
- Execute ALU operation between the external data and the internal data and move the results out of the system.
- Execute ALU operation between two internal data and move the results into the internal register(or memory).
- Execute ALU operation between two internal data and move the results out of the IRAM system.

The above operation can be useful and effective for the conventional arithmetic executions as listed below.

- Read two data from the memory and execute an arithmetic operation and write the result to the memory.
- Read one data from the memory "A" and the other from the memory "B" and execute an arithmetic operation and write the result to the memory "B" or "C".
- Read one data from the memory "A" and the other from the memory "B" and execute an arithmetic operation and use the result for additional arithmetic operations.

We can categorize the external I/O instructions as listed below based on the previous considerations.

- Simple data I/O instructions without internal DSP operation(Type **A**)
- Computational Data I/O instructions without internal DSP operation(Type **B**)
- Simple data I/O instructions with internal DSP operation(by using the internal instruction memory)(Type **C**)
- Computational Data I/O instructions with internal DSP operation(by using the internal instruction memory)(Type **D**)
- Internal DSP operation only by using the internal instruction memory(Type **E**)

LOAD(STORE) instructions move data between the memory mapped registers and DRAM. Arithmetic instructions execute the arithmetic operations including MAC, ALU and Barrel shifter. I/O instructions controls the data flows in I/O pins of chip, especially, the instructions for the internal decoder. MOVE instructions move data between the memory mapped registers. And the MEM operations move data between the DRAM blocks. Since we use memory-mapped registers to simplify the decoder block, RISC(Reduced Instruction Set Computer) type instructions are used and can be efficiently adopted to the data-intensive signal processing. The five formats of instruction can be simplified as shown in Figure 8.

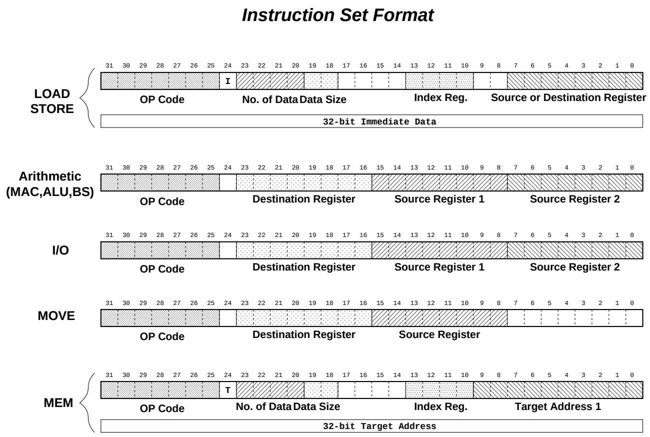


Figure 8: Formats of Instructions for DSP core and External Decoder

For the internal DSP core, four types of instructions are used. Four types of instructions are summarized below.

- LOAD/STORE instruction : Load data from DRAM to data SRAM or registers, or Store data from data SRAM or registers to DRAM
- MOVE instruction : Move data from data SRAM or registers to data SRAM or registers
- Arithmetic instruction : MAC or ALU or Barrel Shifter operation
- MEM instruction : Data movement between DRAM blocks

3.3.1 Internal Instruction Sequence Example

From the algorithm of DCT and the instruction set, we can summarize the instruction sequence of 8-point DCT as shown in Table 4. The number of clocks to execute the 8×8 block 2D-DCT is 448. The required frequency of operation to execute 2D-DCT of moving picture in MP@ML MPEG2 is 72.6MHz.

Using the above instruction sets, we can calculate the required clock frequency to execute the DCT(or Inverse DCT) operation. The required clock frequencies and the sizes of buffer memory for one color at various level of MPEG are shown in Table 5.

Table 4: Instruction Sequences of 8-point 1-line DCT

Instruction	Operation
LOAD R[0],MEM[0][0]	Data loading
LOAD R[1],MEM[0][1]	Data loading
LOAD R[2],MEM[0][2]	Data loading
LOAD ...	Data loading
LOAD R[63],MEM[7][7]	Data loading
ADDS R[64],R[0],R[7]	Add and 1-bit shift
SUBS R[65],R[0],R[7]	Sub and 1-bit shift
ADDS R[66],R[1],R[6]	Add and 1-bit shift
SUBS R[67],R[1],R[6]	Sub and 1-bit shift
ADDS R[68],R[2],R[5]	Add and 1-bit shift
SUBS R[69],R[2],R[5]	Sub and 1-bit shift
ADDS R[70],R[3],R[4]	Add and 1-bit shift
SUBS R[71],R[3],R[4]	Sub and 1-bit shift
MUL ACC,c_0,R[64]	Mul to Accm
MAC ACC,c_0,R[66]	Mul and Accm
MAC ACC,c_0,R[68]	Mul and Accm
MAC ACC,c_0,R[70]	Mul and Accm
MOV R[0],ACCM	Move ACCM to R[0]
MUL ACC,c_2,R[64]	Mul to Accm
MAC ACC,c_6,R[66]	Mul and Accm
MAC ACC,-c_6,R[68]	Mul and Accm
MAC ACC,-c_2,R[70]	Mul and Accm
MOV R[2],ACCM	Move ACCM to R[2]
...	...
MOV R[5],ACCM	Move ACCM to R[5]
MUL ACC,c_7,R[65]	Mul to Accm
MAC ACC,-c_5,R[67]	Mul and Accm
MAC ACC,c_3,R[69]	Mul and Accm
MAC ACC,-c_1,R[71]	Mul and Accm
MOV R[7],ACCM	Move ACCM to R[7]

3.4 Design of Sub-block

We are currently designing all the sub-blocks as they operate at 100MHz with 2V supply voltage. The CMOS process to be used is 0.35μm 4-metal process(Table 3). We design the data-path except the DRAM controller and decoder with full custom approach to get high performance at low V_{CC}. The data-paths are simulated by Verilog-XL simulator and simulated by HSPICE to get the exact timing verifications. The layout is being performed by Cadence layout tool. The DRAM controller and decoder will be synthesized by standard cells.

3.4.1 Design of SRAM and ROM

In our IRAM system, there are two types of SRAM exist. One is instruction SRAM and the other is data SRAM. Two types of SRAM must operate at 100MHz to get single cycle instruction fetch and data **read** or **write**. For data SRAM, two data can be read or written at the same time to support single cycle operand fetch. The block diagrams of instruction SRAM and data SRAM are shown in Figure 9, 10 and the circuits of two types cell are shown in Figure 11.

Table 5: Required Memory Size and Clock Speed for DCT in MPEG

Resolution	DRAM size	I/O Freq.	DSP Freq.
Low	405.6KB	3.04MHz	21.3MHz
Main	1.4MB	10.4MHz	72.6MHz
High-1440	6.3MB	46.7MHz	327MHz
High	8.3MB	62.2MHz	435MHz

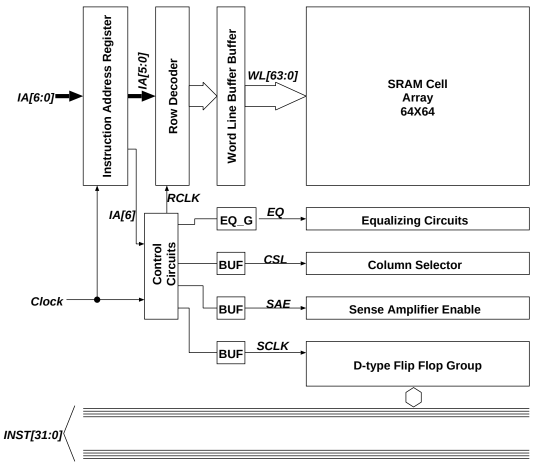


Figure 9: Block Diagram of Instruction Memory(SRAM)

The control signal generator is carefully designed to get fast fetch from instruction memory or data memory. The block diagram of control signal generator is shown in the Figure 12.

To fetch data or instruction in a single cycle, dynamic circuits are used in the row decoder and bit-line pre-charge circuits.

Similarly, ROM is being designed using dynamic circuits to get single cycle READ.

3.4.2 Design of DRAM Block

The access times of DRAM are represented mainly two items, RAS access time and CAS access time. RAS access time is larger than CAS access time and is expressed in (4).

$$t_{RAC} = \tau_{RA} + \tau_{DEC} + \tau_{WL} + \tau_S + \tau_{DB} + \tau_{OUT} \qquad (4)$$

τ_{RA} : Row address set-up time
τ_{DEC} : Decoding time
τ_{WL} : Word line charging time
τ_S : Sensing time
τ_{DB} : Data Bus sensing time
τ_{OUT} : Data output time

The target DRAM will be a synchronous type to get higher speed and be synchronized with DSP core. However, we are designing EDO type DRAM in the test pattern of our IRAM. The DRAM design will be carefully executed by considering below items.

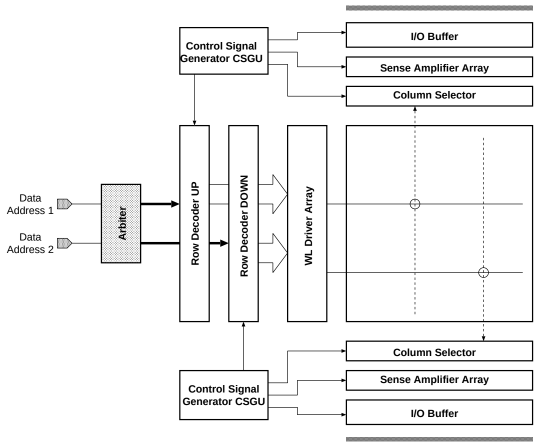
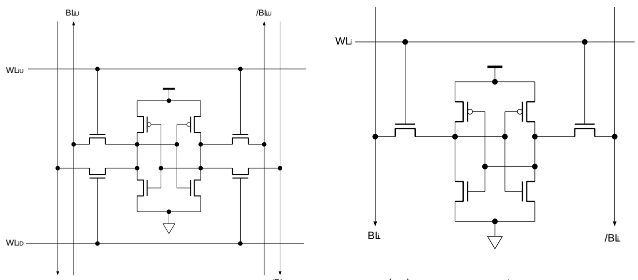


Figure 10: Block Diagram of Data SRAM



(a) 8-Tr SRAM Cell

(b) 6-Tr SRAM Cell

Figure 11: Full CMOS SRAM Cell for IRAM

- Wide bus can be connected to DRAM directly to get high bandwidth. As shown in the analysis of DCT in MPEG, there are four DRAM banks which can be operated independently. And, the width of memory bus must not be the bottleneck of the computation. Thus, we will use wide memory bus.

- Resistance of word line in DRAM circuits can be lowered by some additional metal lines. The resistance and capacitance of word line in DRAM determine the delay time of τ_{WL} in (4).

- Row and column addresses of DRAM can be applied in the same time using wide address bus. In IRAM system, since there are no limitation in the number of address pins, the address multiplexing is not necessary. Thus, row address and column address can be applied in the same time. Cell plate biasing can be used to reduce power dissipation in the IRAM system.

Another design point of DRAM for embedded system is how many banks are generated by dividing the whole DRAM system. We will study the DRAM I/O speed and power dissipation to determine the size of bank and the sub-bank.
The block diagram of EDO-type DRAM which will be inserted into the test pattern of our IRAM system is shown in the Figure 13. The shaded areas indicate the regions which are described in the previous three items. One of the them, "A" indicates the block to execute **WRITE After READ**

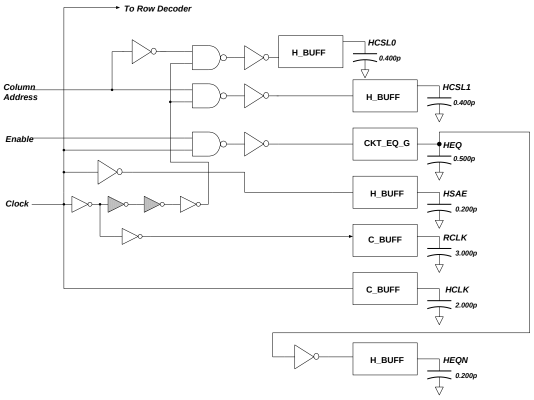


Figure 12: Control Signal Generator

operation. "WRITE After READ" operation can be used effectively in the replacement of data in same memory location. For the DCT application, "WRITE After READ" operation can reduce the size of memory by 25% in use as shown in Figure 14 and it simplifies a DRAM controller.

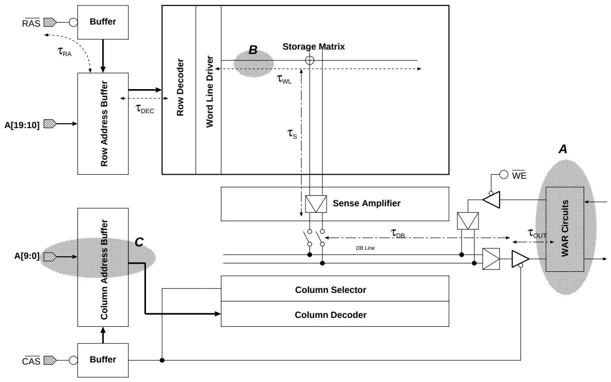


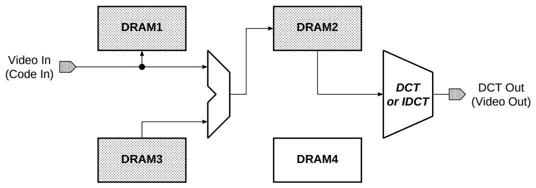
Figure 13: Block Diagram of DRAM in EML System

The schematic diagram of "A" is shown in detail in the Figure 15. This block operates according to the truth table in Table 6. When DRAM operates in the **WAR** mode, the signals *WRITE In* and *DRAM Read* are equal, the states of *READ Out* and *DRAM Write* do not change. But, if *WRITE In* and *DRAM Read* differ, *READ Out* and *DRAM Write* are reserved.

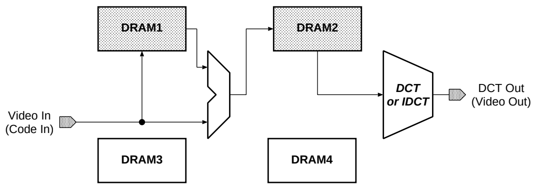
Table 6: Truth table of **WRITE After READ** Circuits

WRITE In	DRAM Read	READ Out	DRAM Write
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

To implement the given operations, circuits for I/O interface of DRAM are shown in Figure 16(a). Connections of the given circuits are changed according to the control signals as shown in Figure 16(b),(c) and (d).



(a) Without "WRITE After READ"



(b) With "WRITE After READ"

Figure 14: Data Flow and DRAM Usage in IRAM for DCT

For the effective use of DRAM as shown in Figure 13, we use the internal DRAM controller. Since, there are two decoders in our IRAM system, it is necessary to control the communication between two decoders and four DRAM blocks. Through DRAM control and status registers(DCSR), the DRAM controller communicates with the decoder for the I/O instructions and the decoder of the internal DSP core. The format of DCSR is shown in Figure 17. In Figure 17, the control and status registers for DRAM controller are divided into two which are corresponding to the external and internal instructions. Each register has been divided into two parts. MSB 16-bit part indicates the request mode of the instructions and LSB 16-bit the status of connection.

3.4.3 Design of Decoder

There are two instruction decoders in our IRAM system. One is used for the external instruction from the master processor and the other is used for the internal instruction from the instruction memory in the system. The architecture of our IRAM system for multimedia applications is shown in Figure 18.

The decoder 1 in the Figure 18 decodes the external instruction and determines the type of the external instruction. And the decoder 1 generates the address of DRAM by using address registers for external access and controls the I/O ALU(including barrel shifter) which execute the arithmetic operation between two data from two different DRAM blocks or a data from one specific DRAM block and an external data. The internal decoder(decoder 2) operates depending on the external instruction as explained in the section 3.3. If the external instruction is type **E**(which is described in section 3.3), the internal decoder controls the whole IRAM system on a chip except I/O ALU unit. The register file, data SRAM and the remaining registers are accessed by memory-mapped register addressing.

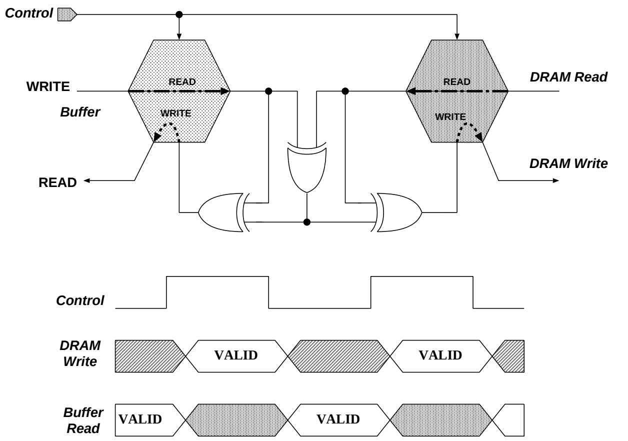
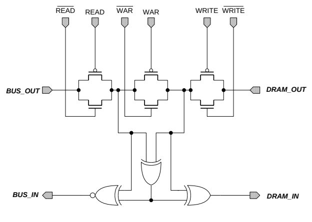
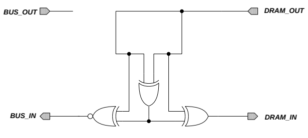


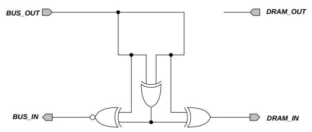
Figure 15: Circuit and Waveform in Data Replacement in a proposed DRAM



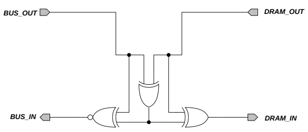
(a) Circuits



(b) READ



(c) WRITE



(d) WRITE After READ

Figure 16: The Circuits for WRITE After READ

4 Future Works

We are designing the IRAM system which can effectively process video signals. But for the efficient IRAM system for multimedia applications, 3-D graphic application should also be considered because the 3-D graphic applications require high computing power and large memory. In addition, for the efficient video signal processing, our research will be continued on how many banks are needed for each DRAM block.

5 Conclusion

We proposed a new IRAM architecture which is optimally focused on the multimedia applications, specifically, DCT operations. The current status is that we are designing the 32-bit 5mm$\times$5mm test IRAM chip using 0.35μm CMOS EML process for embedded memory logic to verify the proposed architecture and circuits. All of the logic elements are

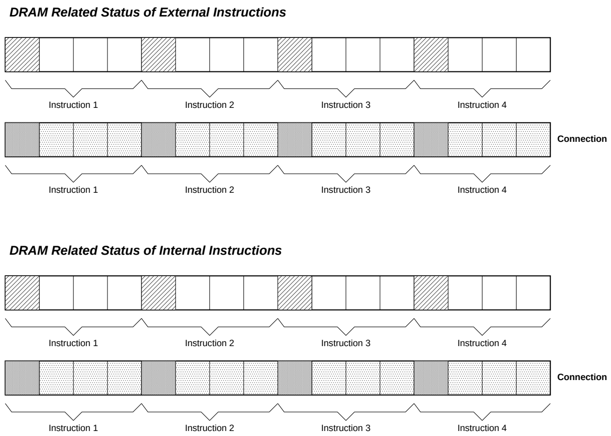


Figure 17: Format of DRAM Control and Status Registers

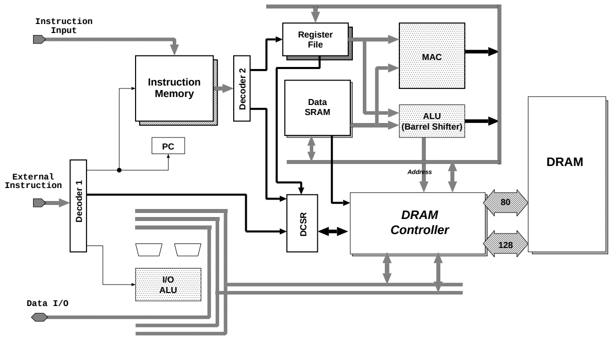


Figure 18: A Proposed Architecture of IRAM for Multimedia Application

designed to be operated at 100MHz with 2V supply voltage. Under the limit of test chip size, the size of DRAM will be minimized to include the required logic circuits. DRAM will be EDO type 4k words. For low power operation, the circuits which are necessary in the arithmetic units are designed by full custom approach. For DRAM circuits, the reduction of the resistance of word line and the removal of the address multiplexing for low power DRAM operation and small access time will be possible (Shaded area "B" and "C" in the Figure 13). The elements of IRAM system are summarized in the Table 7.

In our proposed architecture, for the moving pictures of 720$\times$480 pixels/frame MP@ML in MPEG2, the required clock frequency is about 72.6MHz and the required frame memory is 1.4MB for one color in DCT operation.

Acknowledgment

This work is supported by SAMSUNG Electronics Co., LTD. We would like to thank Sang-Bong Park in SAMSUNG for the preparation of the simulation conditions. Also, we are grateful to Changhyun Kim for the advice on the DRAM architecture and the circuit design.

Table 7: Summary of the IRAM system

Elements	Operation	Remarks
Decoder	Instruction Decoding	RISC type
DRAM	Storage	4 block, WAR Op.,
SRAM	Fast Storage	Instruction and Data
MAC	Multiplication and Accumulation	2V 100MHz
ALU	Arithmetic Op	1-bit Shift After Operation
DLL	Synchronization	Digital
ROM	Storage of Constant	Single cycle operation

References

[1] B. Prince, *Semiconductor Memories:A Handbook of Design, Manufacture, and Application*. Chicherster, England: John Wiley & Sons, 1996.

[2] *ADSP-21060 SHARC Super Havard Architecture Computer*. Norwood,MA, 1993.

[3] D. Patterson and et al., "Intelligent RAM(IRAM):Chips that Remember and Compute," in *ISSCC Digest of Technical Papers*, pp. 224–226, Feb. 1996.

[4] K. Murakami, S. Shirakawa, and H. Miyajima, "Parallel Processing RAM Chip with 256Mb DRAM and Quad Processors," in *ISSCC Digest of Technical Papers*, pp. 228–229, Feb. 1997.

[5] T. Shimizu and et al., "A Multimedia 32b RISC Microprocessor with 16Mb DRAM," in *ISSCC Digest of Technical Papers*, pp. 216–217, Feb. 1996.

[6] D. A. Draper and et al., "An X86 Microprocessor with Multimedia Extensions," in *ISSCC Digest of Technical Papers*, pp. 172–173, Feb. 1996.

[7] B. A. Gieseke and et al., "A 600MHz Superscalar RISC Microprocessor with Out-Of-Order Execution," in *ISSCC Digest of Technical Papers*, pp. 176–177, Feb. 1997.

[8] M. R. Choudhury and J. S. Miller, "A 300MHz CMOS Microprocessor with Multi-Media Technology," in *ISSCC Digest of Technical Papers*, pp. 170–171, Feb. 1997.

[9] C.-H. Kim and et al., "A 32-Bank 1Gb DRAM with 1GB/s Bandwidth," in *ISSCC Digest of Technical Papers*, pp. 378–379, Feb. 1996.

[10] A. Hatakeyama and et al., "A 256Mb SDRAM Using a Register-Controlled Digital DLL," in *ISSCC Digest of Technical Papers*, pp. 72–73, Feb. 1997.

[11] S. Tanoi and et al., "On-Wafer BIST of a 200Gb/s Failed-Bit Search for 1Gb DRAM," in *ISSCC Digest of Technical Papers*, pp. 70–71, Feb. 1997.

[12] A. Inoue and et al., "A 4.1ns Compact 54×54b Multiplier Utilizing Sign Select Booth Encoders," in *ISSCC Digest of Technical Papers*, pp. 416–417, Feb. 1997.

[13] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.