A DRAM Compiler Algorithm for High Performance VLSI Embedded Memories

A. G. Eldin
Electrical Engineering Department,
The University of Toledo
Toledo, Ohio, 43606

Abstract - **In many applications, the limited density of the embedded SRAM does not allow integrating the memory on the same chip with other logic and functional blocks. In such cases, the embedded DRAM provides the optimum combination of very high density, low power and high performance. For ASIC's to take full advantage of this design strategy, an efficient and highly reliable DRAM compiler must be used. The embedded DRAM architecture, cell and peripheral circuit design considerations and the algorithm of a high performance memory compiler are presented .**

1 Introduction

The storage functions of a digital system can be implemented by one of two approaches. The first approach is illustrated in Fig. 1, where standard off-the-shelf memory chips are used. Chip A contains the logic functions whereas chip B provides the memory. This approach is generally adopted if the application does not justify the cost of developing an ASIC chip while the logic and processing functions can be economically and conveniently implemented by off-the-shelf components or semi-custom design using Gate Arrays that do not usually contain or enable the implementation of any sizable storage function efficiently. Additionally, the use of the off-the-shelf memory chips is the only choice in cases where the size of the

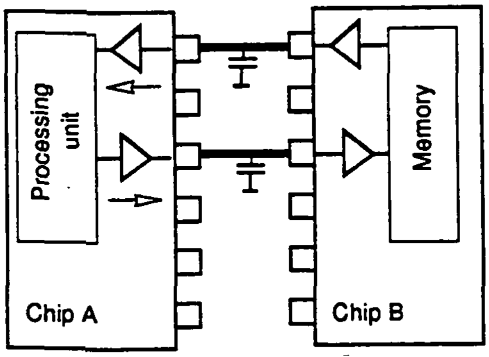


Figure 1: Additional delay due to chip capacitance

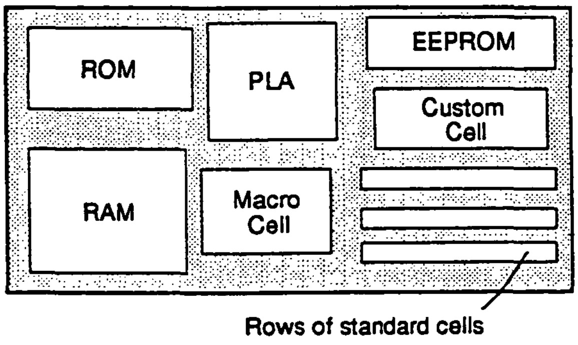


Figure 2: ASIC chip with embedded memory modules

required memory is too large to be integrated with the logic and processing functions on the same ASIC chip. In this approach, there is an additional 40% delay in accessing the memory due to driving large off chip capacitive loads. The second approach is illustrated in Fig. 2 where the memory and logic functions are integrated on the same chip. Embedding memories increases the system speed by eliminating the Input/Output delays resulting from driving large off chip loads. Embedded memories have larger memory bandwidth (*number of bits accessed per second*) as there is no restriction on the number of I/O pins due to packaging requirements. This approach also increases the level of system integration. It provides lower power dissipation and system cost. To take advantage of the embedded memory approach, a reliable and efficient memory compiler (Fig. 3) takes the user's specifications such as type, size, organization, speed and aspect ratio and provides the verified design including the mask data for fabrication, the circuit model for accurate simulation, the logic model, delay parameters and logic symbol for logic and functional simulation and the layout boundary, input/output and signal tables for automatic placement and routing. Embedded/read write Random Access Memories are either static (SRAM) or dynamic (DRAM). A comparison

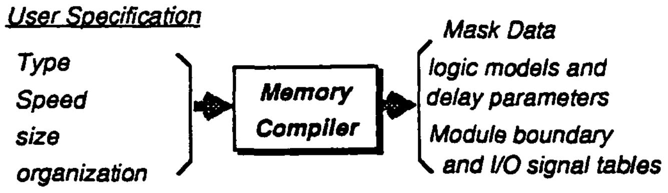

User Specification
Type
Speed
size
organization
Memory Compiler
Mask Data
logic models and delay parameters
Module boundary and I/O signal tables

Figure 3: Memory compiler

between the embedded SRAM using the 6-Transistor (6-T) cell and the 1-Transistor (1-T) cell DRAM shows that the DRAM is approximately four times denser than SRAM while the SRAM is twice as fast as the DRAM [1]. The density advantage of the DRAM is due to the smaller cell size. The DRAM is inherently slower because of the charge sensing of the 1-T cell that requires a more critical timing to ensure that the different phases of the read/write cycle are completed in the proper sequential order. The embedded DRAM is critical for implementing such applications in a cost effective way. There are major differences between embedded DRAM blocks and standard off-the-shelf DRAM chips in terms of fabrication, memory size and organization, design strategy, testing, yield enhancement and applications. The fabrication of standard DRAM chips uses a process that is tailored to provide the maximum number of bits per chip. It offers limited choices in the organization (number of words x number of bits per word). The process provides either the trench or stacked capacitor DRAM cell which are three dimensional structures to realize the maximum capacitance per unit area of the chip. The increase in the process complexity is justified because of the very large production volume of such devices. At the present time, embedded DRAM's are fabricated using standard CMOS process. Three dimensional capacitors are not readily available and therefore the cell area of the embedded DRAM is significantly larger than that of the standard DRAM. This is the reason that standard DRAM chips can have a capacity of up to 25 In this paper, The embedded DRAM design and the compiler's synthesis, verification and characterization algorithms are presented.

2 Embedded DRAM Design

The 1-T DRAM cell is shown in Fig. 4. The storage capacitance Cs = 50 fF. The cell is placed in an N-well to reduce the leakage and alpha particle induced soft error rate. The access transistor is a P-channel transistor. The maximum ratio of the bit line parasitic capacitance Cb to the storage capacitance is (Cb/Cs) = 6.5 which provides a minimum

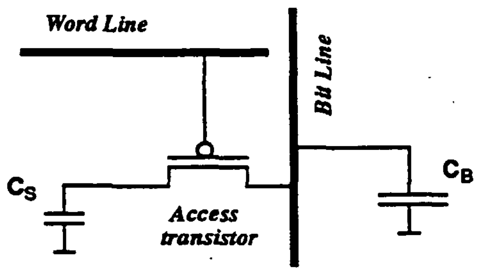


Figure 4: 1-T DRAM Cell

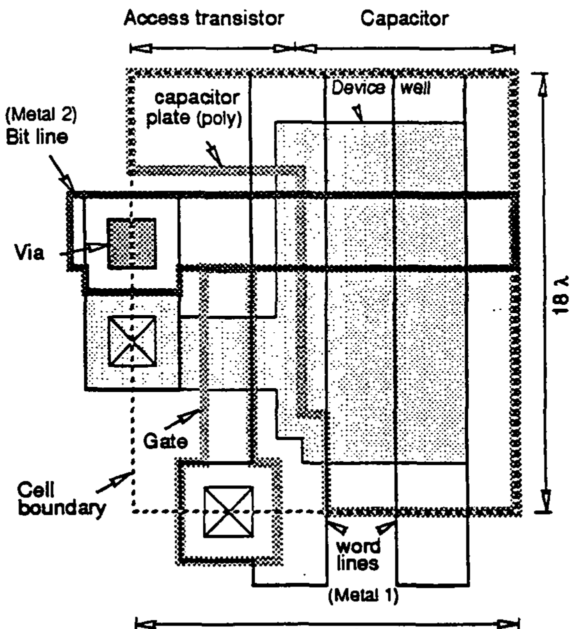


Figure 5: 1-T DRAM Cell layout

sense signal on the bit line of 200 mv. This ensures reliable fast operation. The capacitor plate (polysilicon) is connected to 0 volts and covers the whole array except where the access transistors are formed. This provides low resistance effective shielding of the storage nodes from different noise sources. The metal lines are made of the second layer of metal (Metal2) to reduce the parasitic bit line capacitance and the word lines use the first layer of metal (Metal1). The layout of the cell is illustrated in Fig. 5. Two cells share one drain contact to the bit line and one poly contact to the word line.

The basic block architecture of the embedded DRAM is shown in Fig. 6. It consists of two arrays. Only one array (when selected) remains connected to the sense amplifiers through the bed. All amplifiers switches are lumped together and assembled at the top of the array to improve the layout symmetry of the sense amplifier around the sense lines. This results in higher sensitivity. A new reference voltage generator design is based on the threshold voltage of the CMOS inverter. The circuit is shown in Fig. 7. The reference voltage generator precharges the bit lines to 3.6 volts through the equalization switches. These switches and

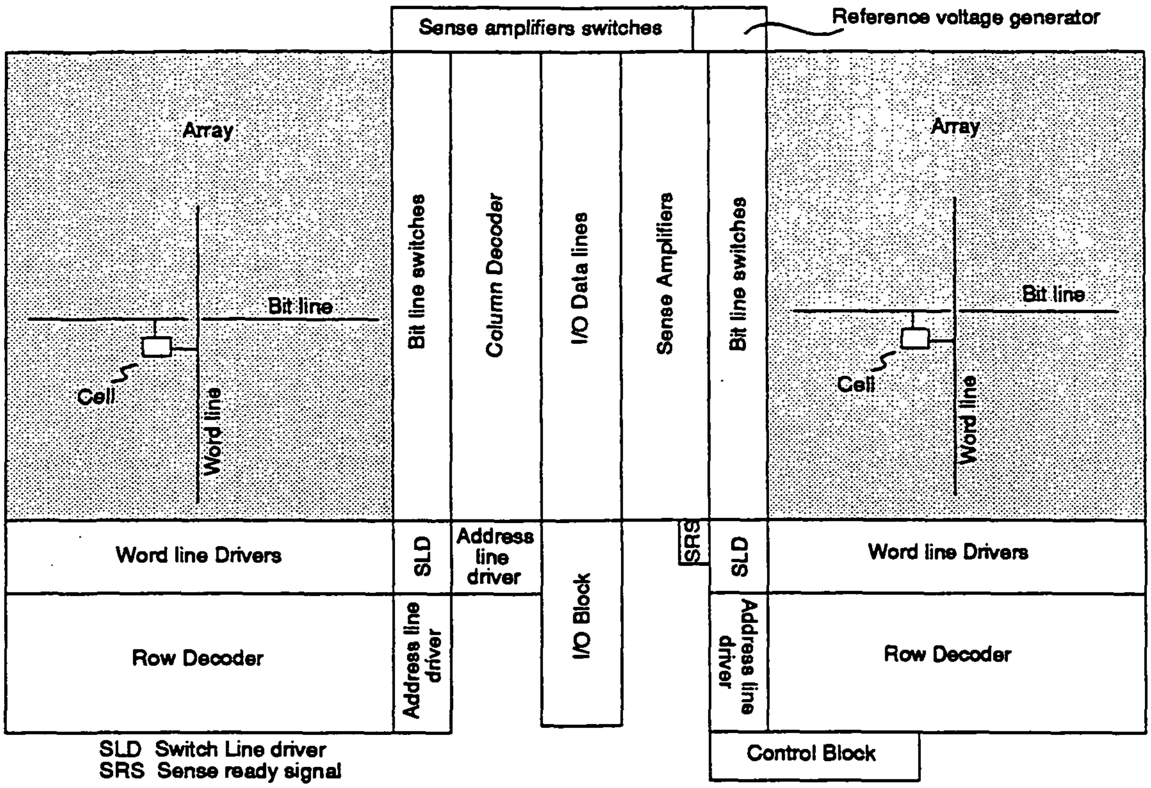


Figure 6: DRAM basic Block diagram

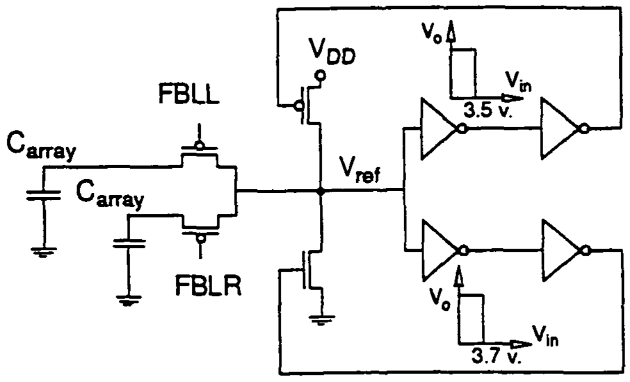


Figure 7: Reference Voltage Generator

the bit line capacitances are represented in Fig. 7 by one lumped transistor and one capacitor for each array. The generator has always at least one array connected to its output. These switches are normally ON except when the array is selected for reading or writing. The output voltage V_{ref} is compared to the threshold voltages of the two inverters which are 3.5 v. and 3.7 v. Either the P-ch or the N-ch transistor will turn on depending on whether V_{ref} is lower than 3.5 v or larger than 3.7 v. to charge or discharge the output to the desired reference level respectively. This circuit accurately produces the desired reference voltage $V_{ref} = 3.6$v. with minimum power dissipation. This reference voltage is approximately the average value of the minimum and maximum voltage levels of the storage nodes which are 2 v. and 5 v. respectively. the input/output circuit uses a gated static latch to realize a 17 ns access time using the 1.5 μm CMOS process. The control block uses self timed circuits to generate the control and timing signals required to ensure that the different phases of the read/write cycle are completed in the proper sequential order for proper DRAM operation.

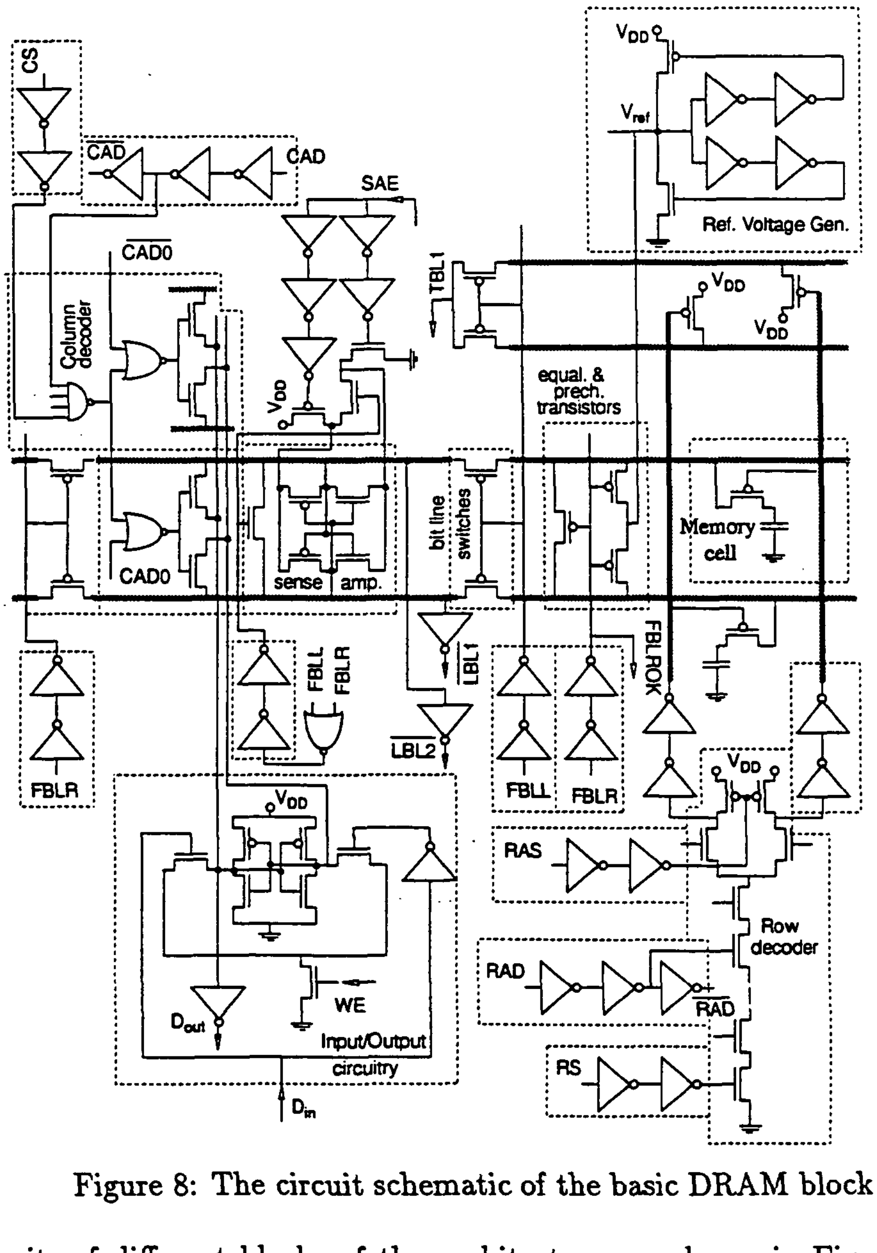


Figure 8: The circuit schematic of the basic DRAM block

The circuits of different blocks of the architecture are shown in Fig. 8 while Fig. 9 illustrates the circuit of the control block. The operation of the DRAM is summarized as follows.

First phase (RAS = 1)

The precharge transistors of the row decoder are turned off. Either FBLR or FBLL is logical 1 depending on the value of MSB being 1 or 0 respectively. This disconnects the precharging and equalization transistors of the selected array and floats the bit lines. The bit line switches of the unselected array are turned off thus disconnecting the bit lines of the array from the common sense amplifiers inputs (sense lines). The floating of the bit lines of the selected array is detected by the FBLROK or FBLLOK depending on selecting the right or left array respectively. Assuming that the right array is selected, FBLROK=1 results in RS=1 which enables the row decoder of the right hand side array. The selected word line potential is pulled down to 0 v. This selects all the cells on that word line. The charge transfer takes place between the cell capacitance and the associated bit line capacitance. The selection of the word line is detected by one of the P-chQQQ transistors on the top column of the array. The turned on transistor on the top of the array causes the sense amplifier enable signal to be SAE=1 after sufficient delay to ensure that the charge sharing mechanism is completed before the sense amplifier is enabled. The turning on of the sense amplifier is detected by a pair of inverters connected to the bottom pair of bit lines. This signal enables the column decoder (CS=1) after making sure that the sense amplifiers have latched to the correct states stored in the memory cells of the selected word line. The column decoder connects the output of the selected sense amplifiers of the I/O data lines through the pass transistors of the decoder. The data of the sense amplifiers overwrite the previous data in the weak latches of the I/O circuitry and in the same time restore the data in the selected memory cells. In the write cycle, the column decoder is enabled immediately after RAS=1. The data to be written (D_{in}) is routed to the selected sense lines and override the much weaker signal from the memory cells. When the sense amplifiers are enabled the D_{in} are written into the selected memory cells.

Second phase (RAS = 0)
The row decoder is disabled (RS=0) and the precharge transistors are turned on to unselect the word line. Once all the access transistors of the memory cells are turned off, the sense amplifiers are disabled (SAE=0). The bit lines are precharged end equalized (FBLR=0) and the bit line switches of the unselected array (left array) are turned on to reconnect the bit lines to the common sense lines. In the same time CS=0 and the column decoder is disabled. This disconnects the I/O data lines from the sense lines. A new cycle can be initiated after the bit lines are equalized at the same reference potential V_{ref}.

3 The DRAM Compiler Algorithm

The compiler performs several functions that include DRAM module synthesis, design verification through DRC and simulation, accurate DRAM module characterization and models for higher level simulation, placement and routing. The correctness of the synthesis algorithm is critical to ensure the proper functionality and high performance of the variable size DRAM while realizing the highest density. This is because the DRAM operation requires accurate timing of sequential control signals and a certain maximum limit of the ratio between the bit line capacitance and the storage capacitances as was detailed in Section 2. The signals delay and the value of the sense signal are strong functions of the different line

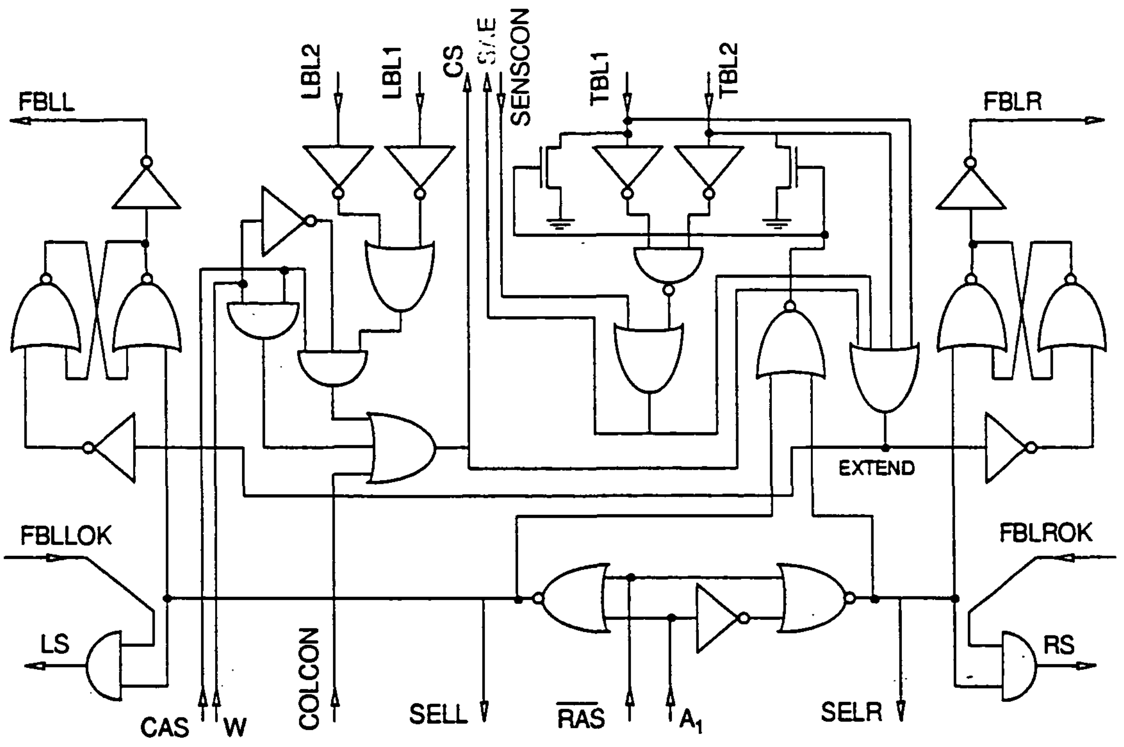


Figure 9: The circuit schematic of the control block

capacitances which in turn heavily depend on the synthesis algorithm.

The algorithm generates DRAM modules that have a size range between 256 bits and 262144 (256 kbits). The maximum limit is based on the assumption that the maximum chip size is 1 cm2 and that the DRAM occupies 80% of the chip area using a 1.2 μm CMOS process. The maximum size also fulfills the storage requirements of most ASIC applications. Below the minimum DRAM size of 256 bits, the DRAM advantage of area savings is diminished and the use of the SRAM is a better choice. The DRAM is built of basic DRAM blocks. Each basic block (Fig. 6) has a maximum of 32768 bits (32 kbits). A block decode logic is used to select on out of a maximum of 16 identical basic blocks. The relative placement of these blocks is determined by the required overall DRAM aspect ratio, the number of generated blocks and the aspect ratio of each block. Table 1 shows the minimum and maximum size and organization parameters of the DRAM module. Table 2 shows the maximum and minimum values of the parameters of the basic DRAM block.

The method of calculating the number of blocks, rows and columns per block, can be explained with reference to Fig. 10. The x and y axes represent the minimum number of basic DRAM blocks required to realize the specified number of words (WORDS) and bits per word (BITS) respectively. The numerical entities in the table indicate the minimum number of identical basic DRAM blocks required for the implementation of a particular DRAM design (module). The number of blocks (BLK) is obtained by multiplying the number of bit blocks (BLK1) by the number of word blocks (BLK2). The word block is 256 words and

Table 2

	Maximum	Minimum	Increment
Number of basic blocks	16	1	1
Number of rows per block	256	4	2
Number of columns per block	128	16	2

Table 1

	Total Size	Total number of words (WORDS)	Total number of bits per word (BITS)
Maximum	262144	262144	1024
Minimum	256	4	1

the bit block is 128 bits. Fig. 10a is used if (BITS > 64) while Fig. 10b for (BITS ≤ 64). For example if the required DRAM module is organized as (280 words $\times$ 300 bits), then BLK1=3, BLK2=2 and the total number of blocks is BLK=6. Each block is organized as (140 words $\times$ 100 bits). The DRAM module is built as shown in Fig. 11 where one block select address line (BS) connects the bit lines either from the blocks labeled (A) or (B) to the I/O data bus. BS is also used to activate the selected blocks. When the number of bits per word (BITS) is larger than 64, no column decoder is required. An other organization example that uses the table of Fig. 10b is (1300 words $\times$ 28 bits) which requires 2 blocks (BLK=2).

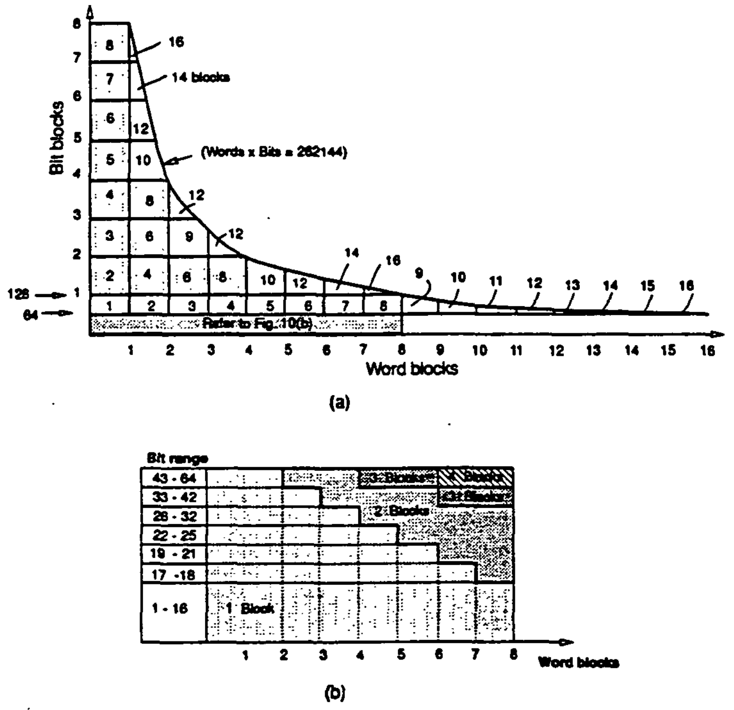


Figure 10: Number of basic blocks for all possible DRAM modules

The algorithm of calculating the number of basic blocks, number of rows (ROWS) and columns (SA) per block is illustrated in the flow chart of Fig. 12. All variables are integers. The algorithm first calculates the minimum number of blocks (BLK1) required to satisfy the

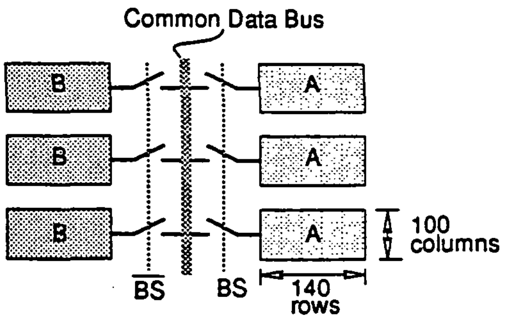


Figure 11: DRAM example module organization

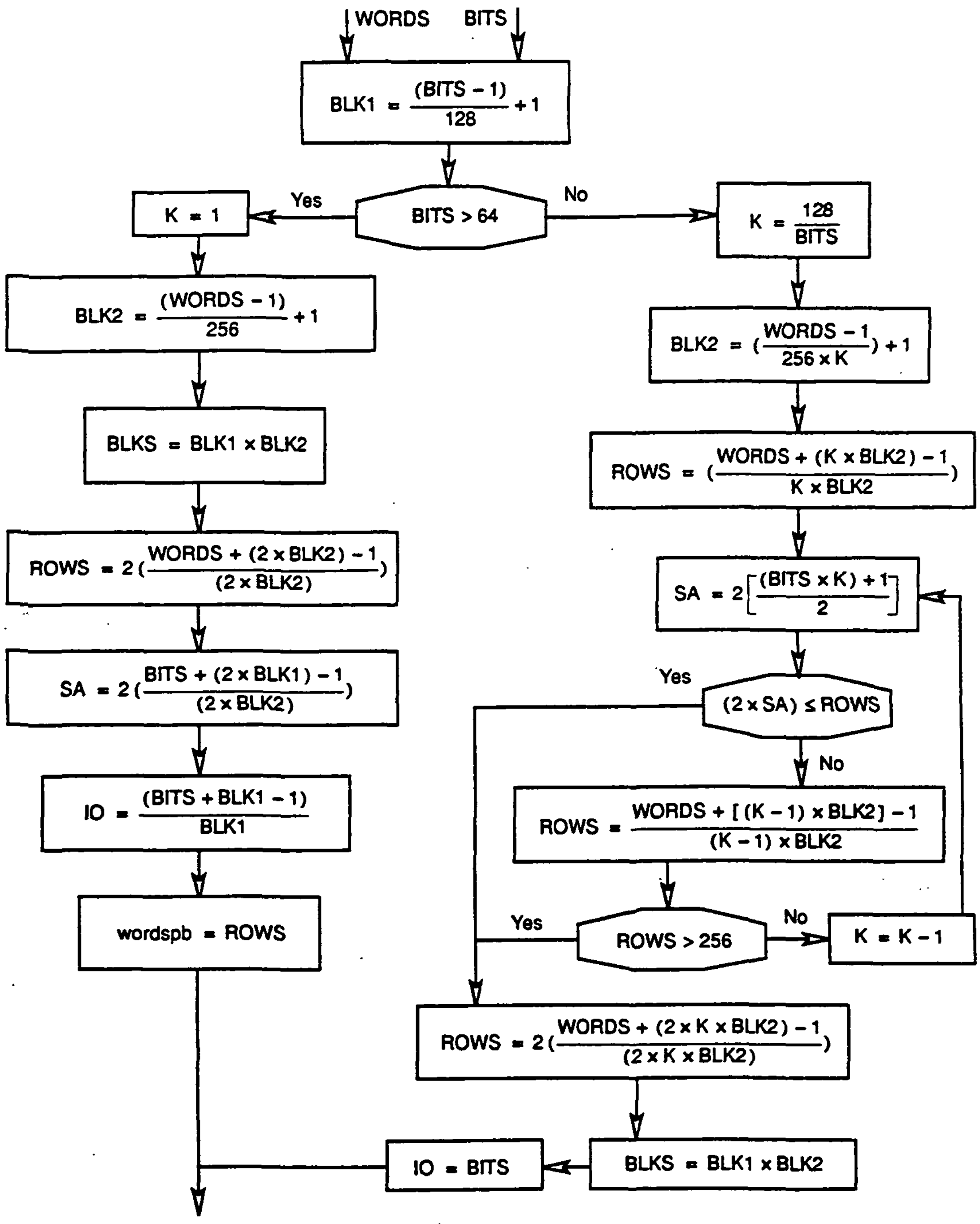
specified number of bits per word (BITS). If (BITS > 64), no column decoder is required (K=1). The minimum number of blocks (BLK2) required to satisfy the specified number of words (WORDS), is calculated. Thus the total number of blocks (BLKS) is determined. The number of rows per block (ROWS), number of sense amplifiers, **which is equal to the number of columns**, per block (SA) and the number of bits per block (IO), **which is equal to the number of I/O data lines**, are calculated. If BITS ≤ 64, the value of k is calculated where

$$k = number\ of\ column\ decoder\ address\ lines\ (n) + 1 \tag{1}$$

This initial value of k gives the maximum number of sense amplifiers and the minimum number of rows per block. To optimize the aspects ratio of the basic block, the algorithm attempts to satisfy the condition ($2 \times$ SA $\leq$ ROWS). This means that the total number of memory cells of one array in the x-direction equals to that in the y-direction. if this condition is satisfied, the array's aspect ratio equals that of the memory cell. This optimizes the speed of operation by providing balanced capacitive loads on the word and bit lines and avoids the generation of DRAM blocks with uncommon dimensions that may be difficult to fit in ASIC chip. This condition is satisfied by iteratively reducing the number of the sense amplifiers while increasing the number of rows. Then number of total blocks is then calculated. A simplified flow chart is shown in Fig. 13 that highlights the basic synthesis and verification steps of the compiler algorithm. The specified size and organization are checked against the values of Table 1 to determine whether they are within the supported limits. The size and number of basic blocks, number of rows and columns per block and the size of the column decoder are determined as explained earlier with reference to Fig. 12. The size of the basic block row decoder in bits (m) is related to the number of rows per block as,

$$2^{m-1} \leq \text{ROWS} \leq 2^{m} \tag{2}$$

The number of I/O data lines and associated circuits (IO) per block is given by ($10 = \frac{SA}{K}$). The sizes of the transistors of the reference generator are scaled according to the total number of bits per block. this for area efficiency and circuit stability. Similarly, the sizes of the sense amplifiers switches, word line drivers, column address drivers and drivers of the bit line switches and equalization and precharge switches are scaled proportionally with the number of sense amplifiers per block. The size of the row address driver is scaled proportional to the



WORDS
BITS
BLK1 = (BITS − 1)/128 + 1
BITS > 64
Yes
No
K = 1
K = 128/BITS
BLK2 = (WORDS − 1)/256 + 1
BLK2 = (WORDS − 1/256 x K) + 1
BLKS = BLK1 x BLK2
ROWS = (WORDS + (K x BLK2) − 1/K x BLK2)
ROWS = 2(WORDS + (2 x BLK2) − 1/(2 x BLK2))
SA = 2⌈(BITS x K) + 1/2⌉
SA = 2(BITS + (2 x BLK1) − 1/(2 x BLK2))
Yes
(2 x SA) ≤ ROWS
No
IO = (BITS + BLK1 − 1)/BLK1
ROWS = WORDS + [(K − 1) x BLK2] − 1/(K − 1) x BLK2
wordspb = ROWS
Yes
ROWS > 256
No
K = K − 1
ROWS = 2(WORDS + (2 x K x BLK2) − 1/(2 x K x BLK2))
IO = BITS
BLKS = BLK1 x BLK2

Figure 12: Algorithm for determining the number of basic blocks and its organization

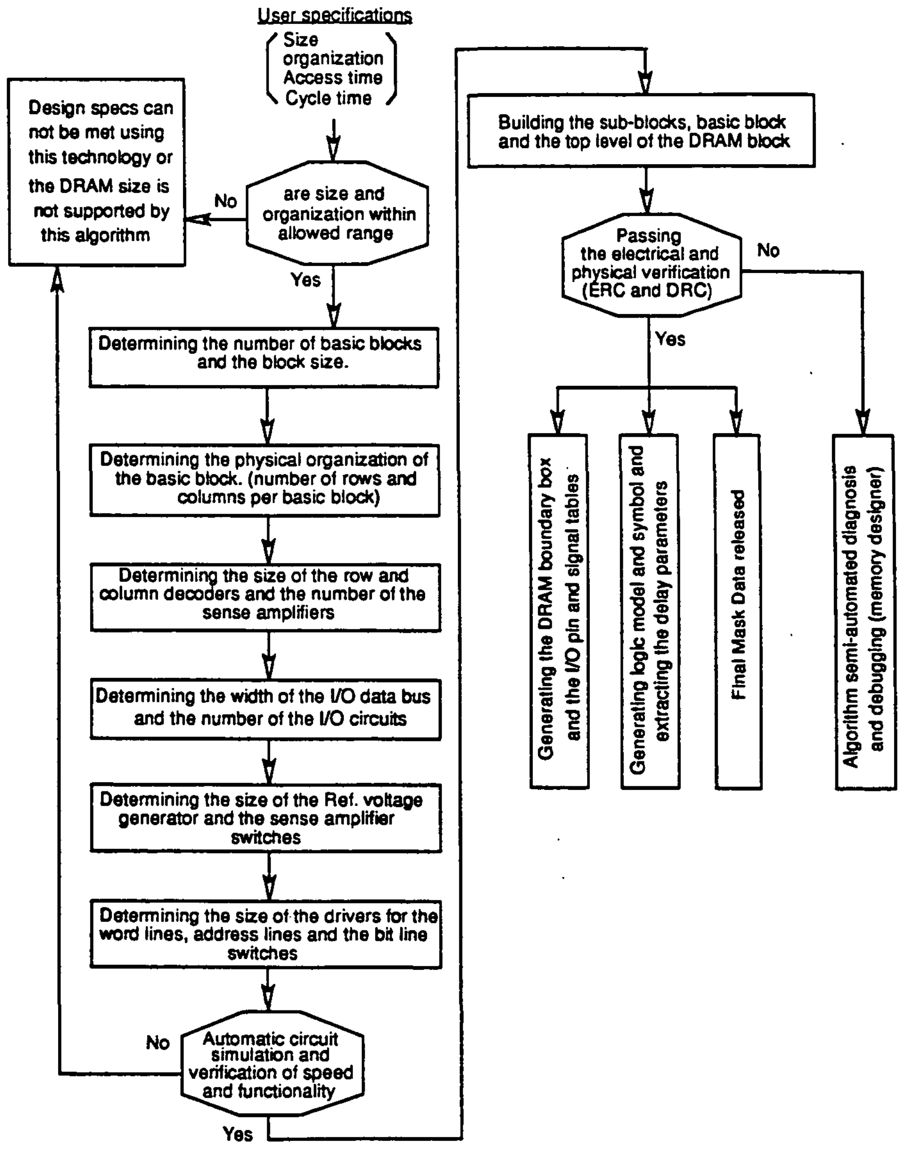

User specifications
Size
organization
Access time
Cycle time
are size and organization within allowed range
No
Design specs can not be met using this technology or the DRAM size is not supported by this algorithm
Yes
Determining the number of basic blocks and the block size.
Determining the physical organization of the basic block. (number of rows and columns per basic block)
Determining the size of the row and column decoders and the number of the sense amplifiers
Determining the width of the I/O data bus and the number of the I/O circuits
Determining the size of the Ref. voltage generator and the sense amplifier switches
Determining the size of the drivers for the word lines, address lines and the bit line switches
Automatic circuit simulation and verification of speed and functionality
No
Yes
Building the sub-blocks, basic block and the top level of the DRAM block
Passing the electrical and physical verification (ERC and DRC)
No
Yes
Generating the DRAM boundary box and the I/O pin and signal tables
Generating logic model and symbol and extracting the delay parameters
Final Mask Data released
Algorithm semi-automated diagnosis and debugging (memory designer)

Figure 13: Summary of the DRAM compiler algorithm

number of rows per array. For each of those variable size circuits, a lower level compiler is used. The scaling of such sub-circuits achieves area efficiency, performance consistency and reliable operation of different DRAM sizes and organizations.

An automatic circuit simulation (using SPICE) and analysis are performed on the resulting DRAM module. The skeleton spice file contains the circuits of Fig. 8 and Fig. 9. The sub-circuits are only extracted once from the layout to reflect parasitic capacitances. The loading effect due to the variable size is modeled using scalable load cells on the word and bit lines. The results of the simulation are automatically analyzed to verify the completion of different phases of the cycle in the proper sequential order and the proper operation of the generated module. It also measures the access and and cycle times and other internal critical delay times. The compiler is capable of generating 20,971,360 different DRAM modules. Therefore, this accurate method of circuit design verification is essential. The following step is to proceed with the building of the sub-blocks, basic DRAM block and the top level of the DRAM module. The generated layout is automatically checked hierarchically for electrical and design rule violations The interior bulk of the cell arrays is excluded to save in processing time and storage requirements. Finally, the mask data are released. The logic model and symbol are generated and delay parameters are accurately extracted from the spice simulation data. The DRAM module boundary box and I/O pin and signal tables are generated. Table 3 summarizes the results obtained using this synthesis algorithm for several DRAM modules. It compares the user's specific specifications with the actually generated modules.

Table 3

User specifications			Generated DRAM Module										efficiency (η) %	Array Aspect Ratio
	Organization											Generated size		
Size (bits)	WORDS	BITS	BLK1	BLK2	BLKS	SA sense amps./block	ROWS rows per block	k	words per block	WORDSG Total words	IO Bits per block	SizeG (bits)		
8073	351	23	1	1	1	46	176	2	352	352	23	8096	99.72	0.87
262144	2048	128	1	8	8	128	256	1	256	2048	128	262144	100	1.67
127167	1311	97	1	6	6	98	220	1	220	1320	97	129360	98.3	1.49
259328	256	1013	8	1	8	128	256	1	256	256	127	262144	98.9	1.67
27675	1025	27	1	2	2	82	172	3	516	1032	27	28208	98.1.	1.59
128000	250	512	4	1	4	128	250	1	250	250	128	128000	100	1.7

The efficiency (η) of the generator is defined as $\eta = (\frac{size}{sizeG} \times 100\%)$ where *size* and *sizeG* are the total specified and generated DRAM module sizes in (bits) respectively. It reflects the compiler's ability to generate area efficient DRAM modules. It is a measure of the compiler's flexibility and the silicon area utilization of the DRAM. The table also lists the aspect ratio of the array for an aspect ratio of the DRAM cell of 1.67.

4 Conclusions

The DRAM compiler presented is capable of generating 20,972,360 different DRAM modules that cover a wide range of sizes, organizations and support very large number of applications.

The use of the described architecture, memory cell, peripheral circuitry, self-timed control signals, variable size sub-circuits, very flexible synthesis algorithm, automatic verification and accurate characterization results in highly area efficient, reliable, dense and high performance DRAM modules.

References

[1] A. G. Eldin "High Speed VLSI SRAMs and DRAMs Evolution and Trends" Proceedings, The International Conference on Microelectronics, Cairo, December 1991.