PASTEL : A Parameterized Memory Characterization System

Kimihiro Ogawa Michinari Kohno Fusako Kitamura
Design Automation Department, System LSI Division, Semiconductor Company, Sony
Email:kimihiro,michi,fusako@saskg.semicon.sony.co.jp

Abstract : PASTEL is a parameterized memory characterization system which extracts the characteristics of ASIC on-chip-memories such as delay, timing and power consumption which are important in LSI logic design. **PASTEL** is a fully-automated process from exact wire-RC extraction through circuit reduction, input vector generation, waveform measurement, data-sheet and library creation. The circuit reduction scheme can reduce the circuit simulation time by 2 order of magnitude while maintaining delay error within $100pSec$ of exact simulation.

1 Introduction

In ASIC-LSI design, the characteristics data for each logic cell (delay, timing constraint and power consumption) are very important since they are used to predict LSI circuit performance. But, it is very time-consuming process to characterize parameterized memories in advance, because they are larger in size and also have many combinations of (word, bit) size. This results in repeated delay calculation for each pair of input_slew and output_load. In addition, characterization is even longer because it is necessary to back-annotate parasitic RCs (resistors and capacitors) to keep the accuracy good enough.

For instance, $1024[word] \times 72[bit]$ memory has 450K Transistors, 3 million Rs and 4 million Cs making it impossible to simulate by Spice-class simulators. Even for a smaller $16[word] \times 36[bit]$ memory, it takes about 1 day to simulate the entire circuit. On the other hand, there is a strong demand from our designers to keep the timing error within $100pSec$ from an exact circuit simulation.

AWE-type RC circuit reducer is widely used [PR90, CN94] to shorten the simulation time, but it can be used only for RC passive circuits and reduce circuits by just around 1 order of magnitude. For our memory case it is indispensable to reduce the number of Transistors in order to use Spice-class simulators. It is a common practice to reduce memories using their regular structure. A methodology to reduce the size of memories was proposed in [WR95], but it needed significant amount of interactive work by designers for standard DRAMs. What we propose is a fully-automated system for parameterized ASIC memories.

We have developed a parameterized memory characterization system, **PASTEL** , which calculates the characteristics of memories under very tight error condition within a reasonable simulation time. We propose a circuit reduction scheme which uses the regular structure of memory. It cuts the non-active portion of the circuit with respect to the signal propagation of interest and merges loads (Transistor, C, R) in order to get the simulation speed-up by more than 2 order of magnitude. A widely used method which replace Transistor by an equivalent C is not so sufficient for our accuracy requirement. Thus, we don't replace them.

PASTEL is a fully-automated consistent process from exact wire-RC extraction of leaf cells (basic unit circuits of memories) through our original circuit reduction, memory synthesis, input vector generation, circuit simulation and waveform measurement to data-sheet and library creation. We are already using **PASTEL** in the production design. When applied to some synchronous SRAMs, we could characterize them in 2 order of magnitude shorter simulation time with only less than $100pSec$ simulation error.

In the remainder of this paper, we explain the system configuration, the Transistor and RC extraction and reduction, the characterization process and the results.

2 System Configuration

PASTEL has 2 major phases, circuit extraction and characterization. In the circuit extraction phase, it constructs netlists from basic leaf cell layouts. The memory synthesizer generates the netlist for the specified (word, bit) combination. Then, in the characterization phase, **PASTEL** creates test vectors for characterization, invokes simulations and characterizes them. It also generates data-sheets and libraries for logic design tools.

2.1 The System Flow of PASTEL

The **PASTEL** system flow is shown in Fig.1. A parameterized memory consists of several kind of leaf cell repetitions and connections among them. First, it extracts Transistors and parasitic RCs of the leaf cells with Quasi-3D layout extractor $Arcadia^{TM}$(Synopsys, Inc.), and then, re-

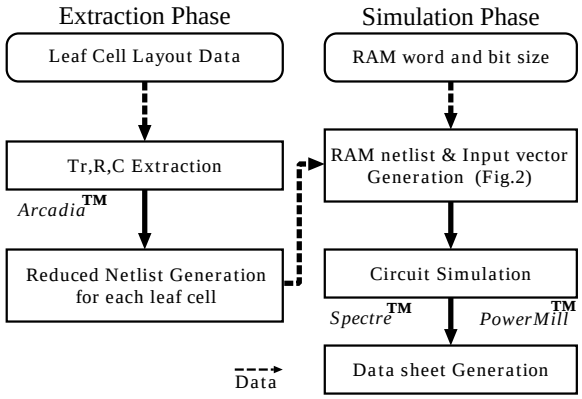

Figure 1: **PASTEL** System Flow

place them by the reduced order equivalent circuits if possible and put them as subcircuits into the top level netlist which the memory synthesizer generates so as to fit for the specified memory size (word, bit). The entire netlist is simulated by $Spectre^{TM}$(Cadence Design Systems, Inc.) ([K95]) or $PowerMill^{TM}$(Synopsys)([CB95]) depending on the purposes(delay, timing constraint, power). Finally, **PASTEL** generates data sheets or libraries for a logic simulator *Blossom*(in-house) , $Verilog^{TM}$(Cadence) and a RTL-level power estimation tool $WattWatcher^{TM}$(Sente, Inc.)([S96])

2.2 Circuit Extraction Phase

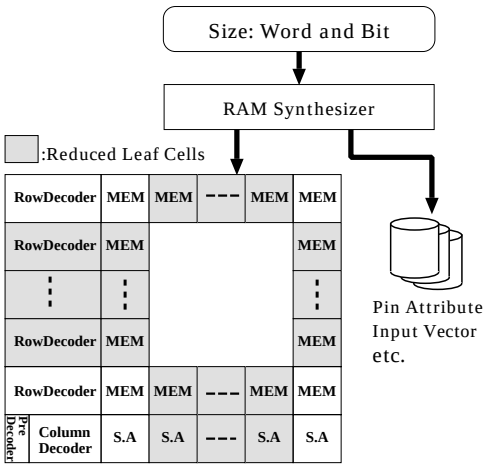

Figure 2: Example of RAM Generation

PASTEL automatically calculates delay, timing constraint and power. For power calculation, the circuits are not reduced in order to obtain enough accuracy. For delay and timing calculations, the circuits are reduced in order to get large speed-up in simulation time. The detailed method of circuit reduction will be described in Sec.3. It approximates one single net segment (rectangle) by a RC π-equivalent circuit and if there are more than one repetitions of the same circuit topology, it merges them to one.

A rough flow of netlist generation is shown in Fig.2. The memory synthesizer generates the entire reduced order netlist of a memory of specified size (word, bit). The way to generate reduced order netlist consists of 2 steps, 1) to remove all portions of the circuit which are not connected to the signal of interest, and 2) to simplify the load transistors and RCs without any significant loss of accuracy. (see Sec.3) And we are not interested in all the signal paths, for instance, for the memory show in Fig.2, only the longest and the shortest signal paths are important for delay and timing, the rest of the circuit are removed.

2.3 Characterization Phase

The memory synthesizer generates simulation netlists for each kind of characterization as well as signal attributes, input vector conditions and other information needed for the characterization as described in Sec.4. **PASTEL** invoke simulators, analyze the resulting waveforms and output the information. Hereafter detail of each process will be described.

2.3.1 Generation of Simulation Netlist

The memory synthesizer generates simulation netlists by combining extracted leaf cell netlists and cutting the portion which has no connection to the signal path. It also outputs physical size of the memory, signal attributes, input vector conditions and other information needed for the characterization.

2.3.2 Generation of Input Vectors

The memory synthesizer automatically generates input vectors and definition files for each kind of characterization. An input vector file is filled with digital values in conjunction with the signal name to be analyzed and its transition status. In a signal definition file, there are described

- Pulse-type: has each one of rising and falling transition in a single cycle.
- Data-type: has either one of rising and falling transition in a single cycle.
- Signal Delay Time, Pulse Width, Initial Conditions.

PASTEL automatically generates necessary test vectors for each simulator referring to input vectors and definition files. The way to generate test vectors will be described in more detail in Sec.4.

2.3.3 Waveform Analysis

When **PASTEL** analyzes simulated waveforms, it reads the signal name and signal transition status from the input file. For instance, for delay characterization, it measures the delay time of clock or data signals and for timing constraint, it searches

the minimum time to get the correct signal on the output pin.

2.3.4 Output of PASTEL

PASTEL prints out data sheets or outputs libraries for *Blossom* , $Verilog^{TM}$ and $WattWatcher^{TM}$. The result characteristics may be a table, the worst case, the average or an piece-wise linear function for the post process.

3 Reduction of Memory Circuits

In this section, we present how to reduce Transistor circuits. For the Transistors which don't propagate the signal of interest, we simplify the load Transistors on the signal path and remove all the rest. In general, the simplification is achieved by a) replacing the load node of Transistor by a simple equivalent linear C or by a little more accurate way which is b) connecting all the rest of the nodes (other than the load node of Transistors) to either V_{dd} or V_{ss} (this can reduce the number of nodes in the circuit). But, neither of them could satisfy our $100pSec$ timing error requirement. So, we decided to use the models described below.

3.1 Reduction of Combinational Circuits

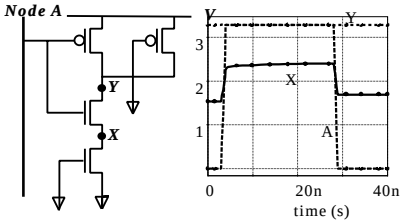

Figure 3: 2input NAND

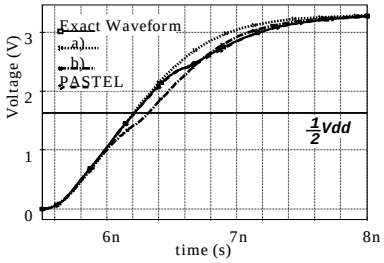

Figure 4: Waveform Comparison for NAND among various methods

An example of load NAND gate which is connected to a signal node A is shown in Fig.3. If we ignore the voltage change of nodes X and Y by adopting approximation a) or b), we noticed that we could not get enough accuracy because of voltage dependent C of the Transistors. We remove all the stages after this NAND, but we have to keep this NAND as it is. In Fig.4, it can be seen that there are more than $150pSec$ time differences around at $1/2V_{dd}$ between b) and the exact one, while there is no significant difference between **PASTEL** and the exact one. a) also differs from the exact one at higher voltage. If there are many load with similar connectivity on one node, effective width of the transistors are summed up to be replace by one large Transistor for NMOS and PMOS respectively.

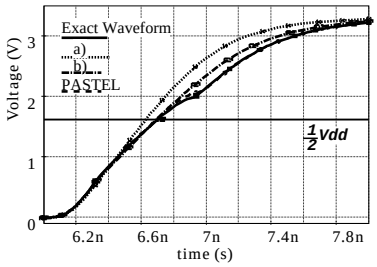

Figure 5: Waveform Comparison for BUFFER among various methods

The same phenomenon can be seen even for a BUFFER. In Fig.5, the voltage point where b) leaves from the exact waveform is higher, but, this can still be explained in the same way. By just looking at this stage, it seems there is no problem, but it affects on the stages on the signal path after this, leading non-acceptable error in delay calculation.

3.2 Reduction of Memory Cells

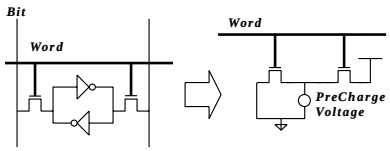

Figure 6: Memory Cell

It can be said the same thing also for memory cells. But, we can use its regular and precharged structure to simplify further. One single memory cell is shown in Fig.6. When a word line is excited , 2 pass Transistors are loaded. It can be assumed that all the bit lines are precharged at $V_{precharge}$ in usual and it is obvious that if the memory cell facing node of a pass Transistor is 1, then that of the other pass Transistor is 0 in spite of memorized data. This suggests we can merge all the load pass Transistors on a row into equivalent one (shown in the right half of Fig.6) by multiplying each effective width of Transistor by number of bit.

Now, we are using 12 types of pre-defined circuit reduction patterns built in Pastel.

3.3 Efficiency of Reduction

In order to check the accuracy and efficiency of **PASTEL** reduction, we compared the results of

non-reduced and reduced circuit simulation. The $Spectre^{TM}$ simulation time of the reduced circuit was only 2% of non-reduced one and the error was only 0.23% in delay time (see Table1).

Table 1: 64word36bit SRAM

Reduction	Yes	No	Difference
Delay(nSec)	2.5478	2.5420	0.23%
Sim Time	6m40s	6h20m22s	-98%
# of Elements	2141	434K	-96%
# of Transistors	1174	18718	-94%

on SUN Ultra1

Table 2: 1024word72bit SRAM

Reduction	Yes	No
# of Transistors	1.3K	455K
# of Resisters	276	3M
# of Capacitors	823	4M
Simulation Time	10min	

on SUN Ultra1

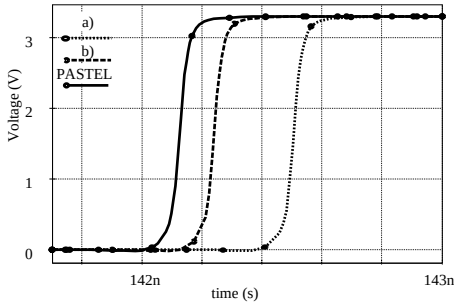

Figure 7: Waveform Comparison of a Signal Delay Path

In Table2 an example of larger memory is shown. It is impossible to simulate this $1024word \times 72bit$ memory by $Spectre^{TM}$, but after **PASTEL** reduction it is simulated in 10 minutes. Adding more explanation, 1) by removing the portions which don't have any connection to the signal net, it can be simulated in around 1 hour and 2) by further merging load circuits of the same topology, it can be simulated in 10 minutes. In Fig.7 it is shown how much error introduced by a) and b) compared to **PASTEL** for the memory shown in Table2. a) has $381pSec$ and b) has $115pSec$ delay differences.

4 Characterization

In this section, the characterization of delay, timing constraints and power by **PASTEL** will be explained.

4.1 Delay

Delay time is the time difference from input signal to output. It is well known that cell delay depends on input slew and output load C, in addition memory delay depends also on data it memorizes. By the regular structure of memory, it is obvious which are the shortest and the longest path. **PASTEL** does measurement only on these shortest and longest paths with changing load Cs and memorized data. For our case , input slew has only negligible effect(several [pSec]) on total delay time.

4.2 Timing Constraint

The definition of memory timing constraint is the same as sequential circuits as shown in Fig.8

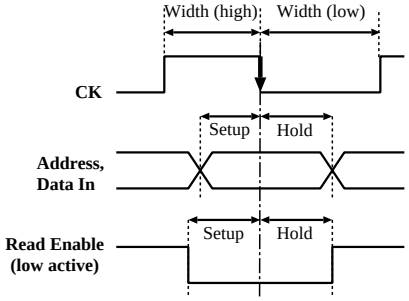

Figure 8: Timing Constraints

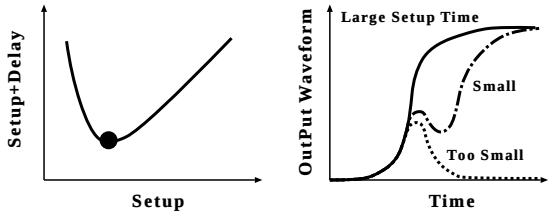

Figure 9: Relation between Setup Time and Delay

For large circuits such as memory, it takes huge amount of simulation time to get the timing constraints if we choose some iterative method, therefore, we had used an analytic method before. The timing constraint can be obtained by first identifying two small signal paths, one from external data-in to the internal latch-data-in and the other is from from an external clock-in to the internal latch-clock-in. The constraint can then be obtained by simulating delay time of both paths, subtract the two delay times and multiply the result by the margin to obtain the final result.

The above calculations can reduce the simulation time very much. However, there is no guarantee that a signal will latch or that a reasonable value is calculated. An example is shown in Fig. 9. If a setup time is close to a critical point, the data may not latch, a nonlinear relation also exists between setup time and delay of that memory from data-in to data-out. If the setup time is not sufficiently large, the waveform at the data-out shows a non-monotonic behavior as show in the right graph while the sum of the setup and delay time increases rapidly as shown in the left graph. This indicates that we have to take a value of setup time as small as possible while maintaining smooth transition of the output waveform. In order to do it, we need to find the minimum point of the

left curve. This can be done through a search process. One more important point here is we need to watch the output waveforms to confirm the circuit is working correctly.

The above search process was implemented in **PASTEL** in order to obtain the constraints correctly. An one dimensional search, Golden section, was used. Since the simulation time has been reduced drastically, the total cpu time required by **PASTEL** is very reasonable. A comparison of **PASTEL** with the former analytical method is shown in table 3

Table 3: Timing Comparison (2048[*word*] × 36[*bit*] SRAM)

Type	Setup (nsec)		Hold (nsec)	
	PASTEL	Analytic	**PASTEL**	Analytic
DataIn	-0.450	0.650	0.0156	1.280
Read	-0.141	0.259	0.2540	0.788
Write	-0.131	0.316	2.2460	0.788
Address	-0.258	1.854	0.1406	0.659
ChipEna.	0.648	2.180	—	—

4.3 Power Consumption

It has not yet been known how to reduce memory for power simulation without significant loss of accuracy, so we don't reduce it. The only one fact we found is that wire Rs have no big effect on power simulation on which our designers require relatively looser accuracy, within 20% error. So, we run $PowerMill^{TM}$ for power simulation to reduce the simulation time by factor of 10 on C-only back-annotated netlists. $PowerMill^{TM}$ suffers only within 5% error in accuracy compared to $Spectre^{TM}$

In order to create the complete polynomial power equation set, **PASTEL** prepares each several test vectors for each operation mode (read, write, standby,,) respectively. It calculates power at each operation mode. For instance, 1-port RAM has the following modes.

1. both of read & write
2. neither of read or write
3. write only
4. read only

Then, it generates the test vectors for both address and data-in which includes 4 patterns such as all bit switching at every clock, all bit stable at every clock, all bit is 0 and all bit is 1. With those test vectors, **PASTEL** measures the coefficients (more than 10 kinds) to be multiplied later by the switching activity data given by *Blossom* or $WattWatcher^{TM}$ to get chip-level power estimation. Notice that each coefficient is function of (word, bit). We'd like to model it with as fewer as possible sample points.

Looking at Fig.10, it can be seen that the relation between number of bit and power is almost linear for each operation mode as well as the relation between number of word and power shown in Fig.11. It is possible to approximate these curves by piece-wise-linear functions with several points.

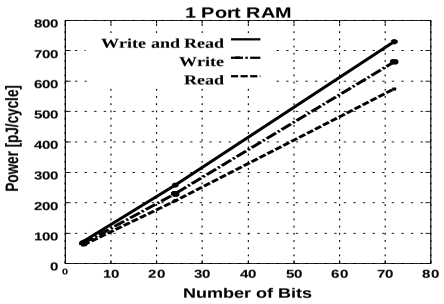

Figure 10: DataBit v.s. Power of 1port RAM

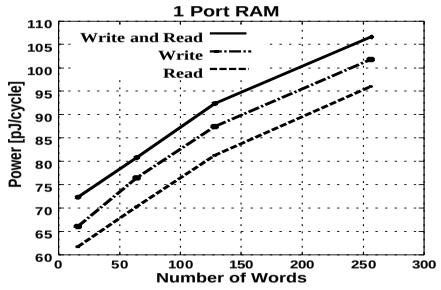

Figure 11: Row v.s. Power of 1port RAM

This implies that fortunately we don't need to simulate the maximum size. Thus, we choose the power sample points as (various words, a few bit) and (a few words, various bits), as shown in Fig.12.

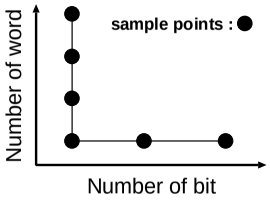

Figure 12: Sample Points for Power Measurement

5 Example of Simulation Time

The simulation time when it is applied to 1-port 1024[*word*] × 4[*bit*] RAM is shown in Table4. It includes all processing time such as simulation and waveform measurement.

6 Example of Accuracy

We compared the delay values between physical measurements and **PASTEL** simulation and presented this in Table5. Including some unknown process deviation of wire RCs (this is quite difficult to know exactly on this test chip), **PASTEL** simulations with calibrated MOS models for this test chip

Table 4: Elapsed Time (1024word:4bit 1port SRAM)

Measurement	Time
Delay (min, max)	30m
Setup	7h
Hold	5h
Width	3h
Power	1h
Input Capacitance	1m
Total	16.5 h

on SUN Ultra1

are within 10% error against the real measurement. This is acceptable for our designers.

Table 5: Delay Comparison

SRAM Word×Bit		Measured (nsec)	PASTEL (nsec)	Error (%)
32×2	d	1.9	1.9	-1.2
128×2	d	2.2	2.2	-1.6
16×1	d	2.0	1.9	-5.7
16×18	d	2.1	2.2	3.0
512×1	d	3.8	3.4	-9.7
32×1	d	2.2	2.0	-7.8
32×18	d	2.7	2.7	0.9
1024×1	d	3.9	3.5	-9.4
64×1	d	2.0	2.0	3.3
64×18	d	3.1	3.2	-4.1
128×2	ds	2.3	2.3	0.1
32×2	s	2.1	2.1	-1.6
128×2	s	2.2	2.3	4.6
128×2		2.2	2.2	0.1

d : d-latch, s : selector

In Table6, we compare *Blossom* simulation (using the library **PASTEL** generates) with $PowerMill^{TM}$ simulation. In the past, to reduce the power simulation time, we have used a simple method which only counts outputs transition time at the gate level power simulation. Our new result is much better than a simple method and keep the error within 15% which satisfies our original requirement of 20%.

Table 6: Power Verification

0.4umCMOS/3.3[V]/20[MHz] 64[*word*]×24[*bit*]

Vec	PowerMill (mW)	PASTEL (mW)	err. (%)	Simple (mW)
1	4.3	3.9	-7.8	8.0
2	5.4	4.7	-12	8.0
3	2.8	2.6	-8.9	8.0

0.4umCMOS/2.2[V]/20[MHz] 64[*word*]×24[*bit*]

Vec	PowerMill (mW)	PASTEL (mW)	err. (%)	Simple (mW)
1	1.8	1.5	-13	3.0
2	2.1	1.8	-13	3.0
3	1.1	1.0	-9.4	3.0

7 Conclusion

We have developed a parametrized memory characterization system, **PASTEL** . It extracts parasitic RCs accurately, reduces circuit size more than 90%, keeps timing accuracy within $100pSec$ and reduces simulation time by more than 2 order of magnitude. It automatically does input vector generation, measurement, data-sheet and library generation. It has considerably improved upon our previous manual and less accurate memory characterization process. Now this is successfully being used in ASIC on-chip-memory characterization.

8 Acknowledgment

The authors thank Mr. Yoshihiko Kinoshita, Mr. Nobuyuki Ishikawa and Mr. Seiya Morinaga for their great contribution on this work from their designer's point of view.

References

[PR90] L.T.Pillage and R.A.Rohrer, Asymptotic Waveform Evaluation for Timing Analysis, IEEE Trans. Computer-Aided Design, vol.9 pp.352-366, Apr. 1990.

[CN94] E.Chiprout and M.S.Nakhla, Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis. Norwell,MA:Kluwer Academic Publishers, 1994

[WR95] W.H.Kao, R.Hamazaki and H.Kikuchi, A Modeling and Circuit Reduction Methodology for Circuit Simulation of DRAM Circuits. IEEE International Workshop on Memory Technology, Design and Testing, pp15-20, Aug. 1995

[CB95] C.X.Huang, B.Zhang, A.Deng and B.Swirski, The Design and Implementation of PowerMill, pp105-109, ISLPD, 1995

[K95] K.S.Kundert, The Designer Guide to Spice & Spectre, Kluwer Academic Publishers, 1995

[S96] Sente,Inc. Sente2.2 Library Development Guide, Dec.2, 1996