2014년 2월 전자공학회 논문지 제 51권 제 2 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 2, February 2014 http://dx.doi.org/10.5573/ieie.2014.51.2.038

논문 2014-51-2-7

기준 메모리를 이용한 메모리 컴파일러 특성화 방법

(Characterization Method of Memory Compiler Using Reference Memories)

신 우 철*, 송 혜 경*, 정 원 영**, 조 경 순***

(Woocheol Shin, Hyekyoung Song, Wonyoung Jung, and Kyeongsoon Cho©)

요 약

본 논문에서는 메모리 컴파일러를 정확하고 빠르게 특성화할 수 있도록 기준 메모리를 기반으로 특성화하는 방법을 제안하였다. 제안한 특성화 방법은 메모리 컴파일러의 정확도를 유지하면서 특성화 시간을 최소화하기 위해 메모리 컴파일러의 타이밍 경향을 분석하고 분석 결과를 토대로 기준 메모리를 선정하고, 메모리간의 경향성을 대변할 수 있도록 모델링하였다. 본 논문에서 제안한 방법론을 검증하기 위하여 130nm에서 개발된 메모리 컴파일러를 제안한 방법을 이용하여 110nm 메모리 컴파일러를 특성화하였다. 이를 통해 생성한 메모리들의 특성과 SPICE를 사용하여 특성화한 결과를 비교하여 메모리 타이밍의 평균 오차율은 ±0.1% 이내였으며 실제 110nm 공정을 사용하여 제작된 메모리 BIST(Built-In Self Test) 테스트 칩으로 기능 검사한 결과, 수율(Yield)이 98.8% 임을 확인하였다. 또한, 180nm 공정을 사용하여 비교한 결과, 수율이 98.3%로 그 유용성을 확인할 수 있었다.

Abstract

This paper proposes a characterization method based on the reference memory to characterize memory compiler quickly and accurately. In order to maintain the accuracy of the memory complier and to minimize characterization time, the proposed method models the trends of the generated memories by selecting the reference memories after analyzing the timing trends of the memory compiler. To validate the proposed method, we characterized the 110nm memory compiler derived from 130nm memroy compiler. The average error rate of the characteristics of the memories generated by the proposed method and SPICE simulation is lower than ±0.1%. Furthermore, we designed memory BIST test chips at 110nm and 180nm processes and the results of the function test show that the yield is 98.8% and 98.3%, respectively. Therefore, the proposed method is useful to characterize the memory compiler.

Keywords : Memory Compiler, Memory, Characterization, SRAM, Embedded Memory

Ⅰ. 서 론

정보 처리 기술의 발달과 애플리케이션 소프트웨어의 증가에 따라 스마트 폰, 게임기, DMB(Digital Multimedia Broadcasting), PMP(Portable Multimedia Player) 등과 같은 기기들이 처리하여 할 정보의 양은 급격히 늘어나고 있으며, 이에 따라 SoC(System on Chip) 내에서 임베디드 메모리의 필요성이 급격히 증가하고 있다. 특히, 스마트 폰이나 게임기 같은 모바일 기기들은 속도에 매우 민감하여 SRAM(Static Random Access Memory)과 같은 임베디드 메모리의 성능에 많은 영향을 받는다. 따라서 임베디드 메모리의 성능을 주어진 공정에 대하여 정확하고 빠르게 특성화(characterization)하는 기술이 필요하다. 여기서 특성화란 해당 IP(Intellectual Property)의 타이밍, 동적 전력, 누설 전력 등의 특성을 라이브러리로 나타내는 것을 의미한다.

* 정회원, ** 평생회원, 동부하이텍(주) TE 팀
(Technology Enabling Team, Dongbu HiTek)
*** 평생회원, 한국외국어대학교 전자정보공학부
(Department of Electronics Engineering, Hankuk University of Foreign Studies)
© Corresponding Author(E-mail: kscho@hufs.ac.kr)
접수일자: 2013년10월3일, 수정완료일: 2014년2월4일

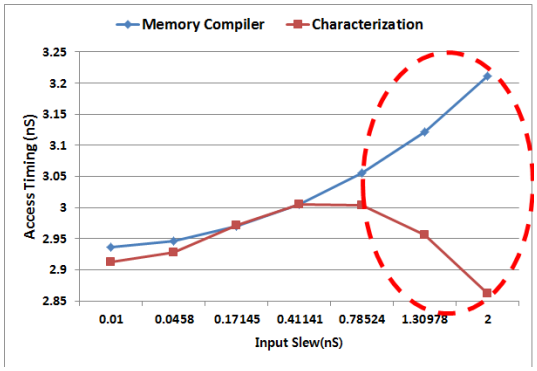

그림 1. 메모리 컴파일러와 SPICE 시뮬레이션간의 SRAM 엑세스 타이밍 차이
Fig. 1. SRAM access timing discrepancy of between memory compiler and SPICE simulation.

일반적으로 임베디드 메모리를 특성화하기 위하여 SPICE 시뮬레이션을 통해 특성화하는 방법과 메모리 컴파일러를 사용하여 특성화하는 방법이 널리 사용되고 있다.[1~3] 전자의 경우, 사용되는 모든 메모리에 대하여 특성화함으로써 그 결과가 정확하나 시간과 CPU core가 많이 필요하게 된다. 반면에 후자의 경우, 특정 leaf cell을 특성화한 후 모델을 만들어 메모리를 특성화함으로써 계산 시간을 줄일 수 있지만 그림 1과 같이 입력 slew가 큰 영역에 SPICE 시뮬레이션을 사용하여 특성화한 엑세스 타이밍과 많은 차이를 보인다.[2]

이와 같은 문제점들을 해결하기 위하여 본 논문에서는 기준 메모리(reference memory)를 기반으로 입력 slew가 작은 영역뿐만 아니라 입력 slew가 큰 영역에서도 많은 메모리들을 짧은 시간 내에 정확하게 특성화할 수 있는 새로운 방법인 AMCM(Advanced Memory Characterization Method)을 제안하고자 한다. 본 논문에서 제안한 방법의 정확도를 검증하기 위하여, 본 논문에서 제안한 AMCM을 사용하여 특성화한 결과와 SPICE 시뮬레이션을 사용하여 특성화한 결과를 비교하였으며, 메모리 BIST(Built-In Self Test) 회로가 적용된 테스트 칩을 110nm 공정과 180nm공정을 사용하여 제작한 후 기능 검사(functional test) 수율을 통해 그 유용성을 확인하였다.

Ⅱ. Advanced Memory Characterization Method

그림 2는 주어진 공정에 대하여 본 논문에서 제안한 AMCM을 이용하여 메모리를 특성화하는 과정을 보여주고 있다.

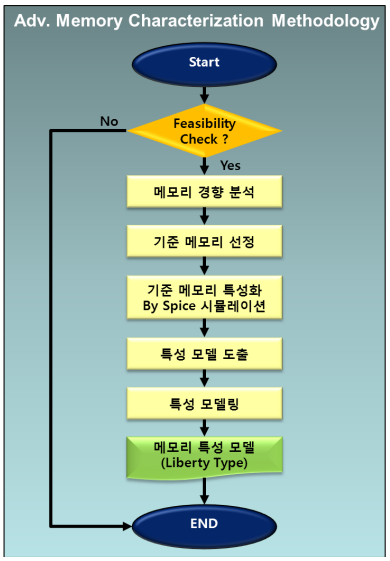

그림 2. AMCM 흐름도
Fig. 2. Flow chart of AMCM.

첫째, 메모리 컴파일러의 활용성을 검증하기 위하여 SPICE 시뮬레이션을 통해 메모리의 동작 여부와 타이밍 등과 같은 기본 특성들을 파악한다.

둘째, 기준 메모리를 선정하기 위하여 주어진 메모리 컴파일러를 이용하여 특성화를 수행할 메모리의 크기별로 엑세스 시간, setup 시간, hold 시간 등과 같은 타이밍에 대한 경향성을 분석한다. 엑세스 시간 vs. 입력 slew, setup 시간 vs. 입력 slew, 그리고 hold 시간 vs. 입력 slew에 대하여 타이밍이 크게 변하는 변곡점에 위치한 크기의 메모리를 기준 메모리로 선정한다.

셋째, 선정된 기준 메모리는 SPICE 시뮬레이션을 사용한 특성화를 수행하여 메모리 특성 값을 추출한다. 기준 메모리에서 추출된 특성 값들을 모델링 프로그램에 이용할 수 있도록 데이터베이스를 구축하며, 이를 이용하여 기준 메모리간의 타이밍 경향을 모델링한다.

넷째, 도출된 모델링 식을 사용하여 기준 메모리 이외의 범위에 있는 메모리들을 특성화한다.

다섯째, AMCM의 최종 결과로서 liberty[5]형태의 타이밍 라이브러리를 생성하게 된다.

이와 같이 함으로써 메모리 컴파일러가 생성하는 모든 메모리를 SPICE 시뮬레이션이 아닌 실험식을 기반으로 특성화를 통해 SPICE 시뮬레이션에 근접하는 정확도를 유지하면서 메모리 특성화에 소요되는 시간을 최소화할 수 있다.

1. Feasibility Check

그림 3은 일반적인 메모리에 대한 구조를 도시하였

다. 위에서 보는 바와 같이 메모리는 bit-cell 배열과 sense amplifier, IO, column decoder, row decoder, control 블록으로 구성되어진다[6].

해당 공정에서 메모리 컴파일러에 대하여 SPICE 시뮬레이션을 통한 활용 가능성 검토는 반드시 필요하다. 메모리의 특성 중 bit-cell 배열에 대한 읽기 특성과 sense amplifier 특성은 메모리의 동작 가능성을 대변한다. 그래서 메모리 컴파일러의 활용 가능성 검토를 위한 방법으로 그림 4와 같이 bit-cell 배열에 대한 읽기 특성과 sense amplifier 특성의 시뮬레이션 결과를 분석하여 bit-cell로부터 출력된 신호가 latch sense amplifier를 구동하기 위한 bit-line과 $\overline{\text{bit-line}}$의 신호의 전위차 △V와 비교하여 충분히 큰지를 검토한다.[3] 특히, 메모리 컴파일러를 다른 공정에 대하여 재사용할 경우, 디자인 룰 간에 문제는 없는지 검토가 진행되어야 하며 이에 대한 타이밍 특성 및 디자인 룰 간의 검토를 통해 파생 공정에 대한 메모리들의 동작에 문제가 없는지도 확인할 수 있다.

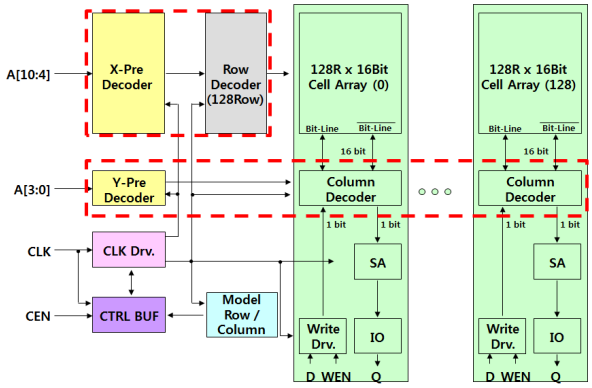

그림 3. SRAM의 기본 구조
Fig. 3. Basic architecture of SRAM.

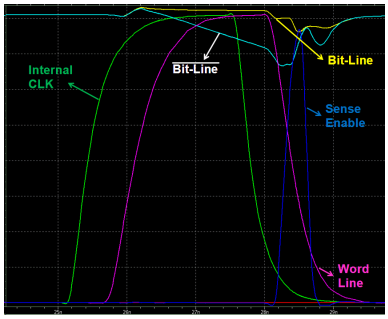

그림 4. Bit-cell 특성과 sense amplifier 특성 분석
Fig. 4. Analysis of bit-cell and sense amplifier characteristics.

2. 메모리 경향 분석 및 기준 메모리 선정

메모리 간의 타이밍 경향성 분석에 앞서 메모리 분석

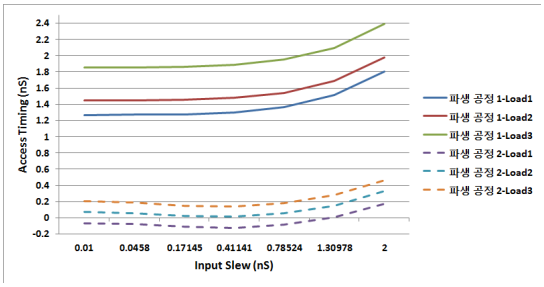

그림 5. 동일 구조 메모리의 공정 간 경향 분석
Fig. 5. Trend analysis of memories with same architecture for each process.

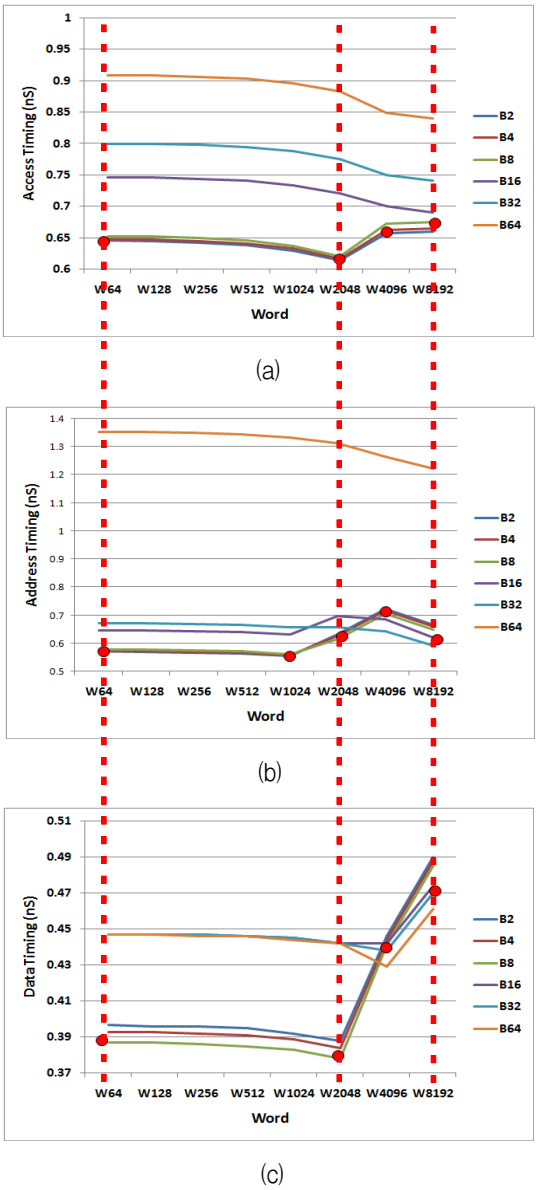

그림 6. (a) Word 변화에 따른 엑세스 타이밍 경향
(b) Word 변화에 따른 어드레스 타이밍 경향
(c) Word 변화에 따른 데이터 타이밍 경향
Fig. 6. (a) Access timing trend for word variation.
(b) Address timing trend for word variation.
(c) Data timing trend for word variation.

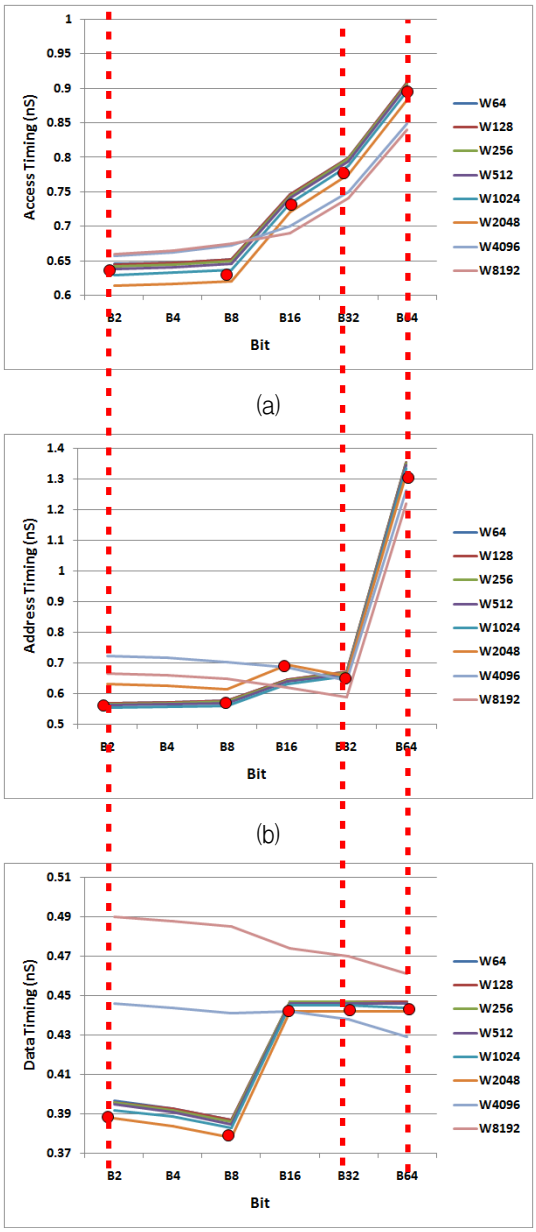

그림 7. (a) 출력 bit 변화에 따른 엑세스 타이밍 경향
(b) 출력 bit 변화에 따른 어드레스 타이밍 경향
(c) 출력 bit 변화에 따른 데이터 타이밍 경향
Fig. 7. (a) Access timing trend for output bit variation.
(b) Address timing trend for output bit variation.
(c) Data timing trend for output bit variation.

의 기준이 필요하며 이는 메모리의 구조에 따라 결정된다. 그림 3을 살펴보면 동일 크기 메모리의 physical word line은 MUX 구조의 변화에 따라 변화하며, row decoder의 구조 역시 변화하게 된다. 또한, bit-line은 MUX 구조와 출력 bit에 의해 변화하며, column decoder 구조도 역시 MUX 구조에 따라 변화함을 알 수 있다. 따라서 메모리의 크기 및 MUX 구조에 따라 메모리의 입력, 출력 타이밍 특성이 변함을 알 수 있다. 결론적으로 메모리의 타이밍 경향성 분석은 MUX 구조별로 word 변화에 따른 특성 분석과 bit 변화에 따른 분석이 이루어져야 한다.

반면, 공정 별 경향은 분석해보면 그림 5의 결과처럼 메모리 구조가 변하지 않은 이상 공정 특성에 따라 메모리의 엑세스 타이밍 경향이 바뀌지 않음을 알 수가 있다. 따라서 구조가 동일하다면 타당성 검토에서 문제가 없을 시, 기존에 분석한 경향 결과를 사용하더라도 문제가 없다.

위의 그림들은 메모리의 word 별, bit 별 각 입력, 출력의 타이밍 특성 경향을 분석하여 얻은 그래프들이다. 그림 6~그림 7에서 보는 바와 같이 각 입력 타이밍, 출력 타이밍에 대한 경향 그래프로부터 타이밍 특성이 바뀌는 지점(원형 점)에 해당하는 여러 개의 메모리를 선정한 후, 각 입력, 출력에 대한 타이밍의 경향을 모두 대변할 수 있는 지점(점선)을 선정한다.

이와 같이 기준 메모리를 선정하여 메모리 특성화할 메모리 수를 줄여 SPICE 시뮬레이션 시간을 최소화할 수 있으며, 경향성을 보장하는 기준 메모리를 특성화함으로써 SPICE 시뮬레이션에 근접한 정확도를 유지할 수 있다.

3. SPICE 시뮬레이션을 이용한 기준 메모리 특성화

메모리의 정확한 특성을 구하기 위해서 RC 성분이 포함한 네트리스트인 DSPF (Detailed Standard Parasitic Format)을 사용하여 특성화한다. 따라서 모든 RC 성분을 포함한 네트리스트를 그대로 사용하여 특성화를 진행하게 되면 많은 시간을 소요하게 된다. 이를 단축하기 위해 타이밍 특성 분석에 필요한 부분만을 선택하여 특성화를 진행하는 작업이 필요하다.[4] 이를

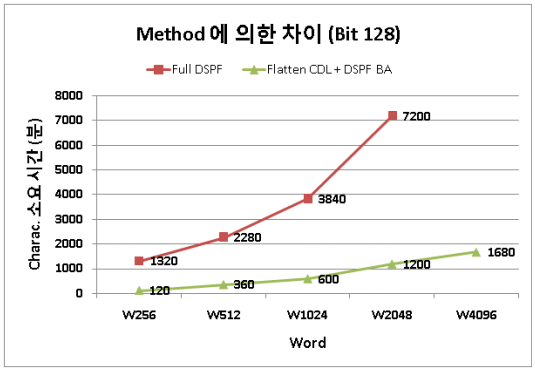

그림 8. Word size(128-bit 고정)에 따른 특성화 시간
Fig. 8. Characterization time for each word size. (128-bit fix).

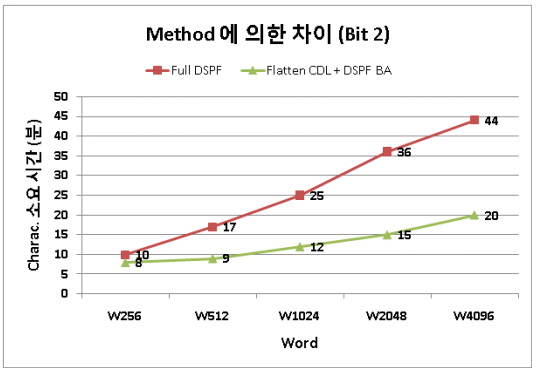

그림 9. Word size(2-bit 고정)에 따른 특성화 시간
Fig. 9. Characterization time for each word size. (2-bit fix).

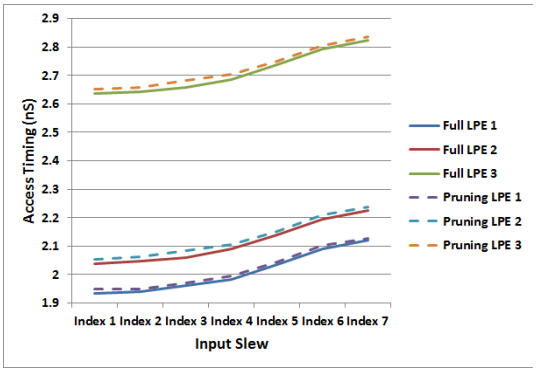

그림 10. 입력 slew에 따른 엑세스 타이밍의 SPICE 시뮬레이션 결과
Fig. 10. Access timing from SPICE simulations over input slew.

pruning작업이라 부른다. 그러나 이 역시 RC성분을 포함한 DSPF를 사용함으로 pruning에 많은 시간이 필요하다. 보다 효율적인 pruning을 위하여 DSPF 대신 CDL (Cadence Design Language) 네트리스트를 통하여 필요한 부분을 pruning하며, pruning된 CDL 네트리스트에 RC 성분을 back annotation 하여 특성화를 진행한다. 그림 8~그림 9는 DSPF를 사용하여 특성화한 시간과 pruning된 CDL 네트리스트에 RC성분을 back annotation 하여 특성화한 시간을 비교하였다.

pruning작업이라 부른다. 그러나 이 역시 RC성분을 포함한 DSPF를 사용함으로 pruning에 많은 시간이 필요하다. 보다 효율적인 pruning을 위하여 DSPF 대신 CDL (Cadence Design Language) 네트리스트를 통하여 필요한 부분을 pruning하며, pruning된 CDL 네트리스트에 RC 성분을 back annotation 하여 특성화를 진행한다. 그림 8~그림 9는 DSPF를 사용하여 특성화한 시간과 pruning된 CDL 네트리스트에 RC성분을 back annotation 하여 특성화한 시간을 비교하였다.

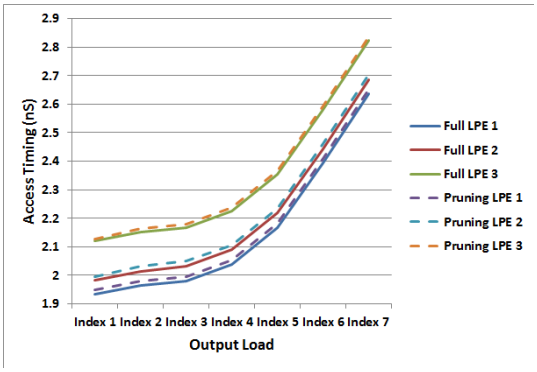

그림 11. 출력 load에 따른 엑세스 타이밍의 SPICE 시뮬레이션 결과
Fig. 11. Access timing from SPICE simulations over output load.

그림 10은 3가지 크기의 메모리에 대하여 입력 slew에 따른 엑세스 타이밍 특성에 대하여 pruning 과 pruning 하지 않은 특성화 간의 차이를 보여 주며, 그림 11은 3가지 크기의 메모리에 대하여 출력 load에 따른 엑세스 타이밍 특성에 대하여 pruning 과 pruning 하지 않은 특성화 간의 차이를 보여 준다. 이와 같이 SPICE를 사용하여 특성화 시간을 줄이기 위해 pruning 할 경우 pruning없이 시뮬레이션 할 때 보다 약 0.2% 더 크나 오차 범위 내에 있다고 볼 수 있다.

4. 메모리의 특성 모델링

앞에서 얻은 데이터를 사용하여 메모리 컴파일러에서 생성되는 기준 메모리 이외의 영역에 존재하는 메모리를 특성화하기 위한 수식을 실험 기반으로 모델링한다. 그림 12와 같이 특성화 수식은 word와 word 간의 가중치를 적용하여 1차 실험 기반 수식을 구한다. 이 식은 word 변동에 따른 기준 메모리 특성을 설명하지만 bit에 따른 변동을 설명할 수 없다. 따라서 bit의 변동에 의한 특성을 구하기 위해 word에 대한 실험 기

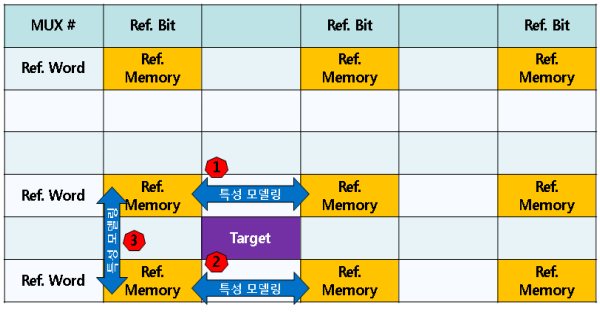

그림 12. 기준이외 메모리에 대한 특성 모델링 알고리즘
Fig. 12. Characteristics modeling algorithm for non-reference memories.

반 수식과 같은 방법으로 bit과 bit간의 가중치를 고려하여 2차원 내삽법(interpolation)을 적용한 실험 기반 수식을 구한다. 이 두 식을 이용하여 기준 메모리 이외 영역에 존재하는 메모리들을 빠르고 정확하게 특성화 할 수 있다.

5. Timing Library 생성

앞에서 서술한 모델링 단계를 거쳐 도출된 모델 값들은 AMCM의 최종 결과로서 대부분의 EDA(Electronic Design Automation) 툴에서 사용이 가능하도록 liberty 형태의 타이밍 라이브러리를 생성하였다. 그림 13은 AMCM을 통해서 생성된 liberty 예제를 보여준다.

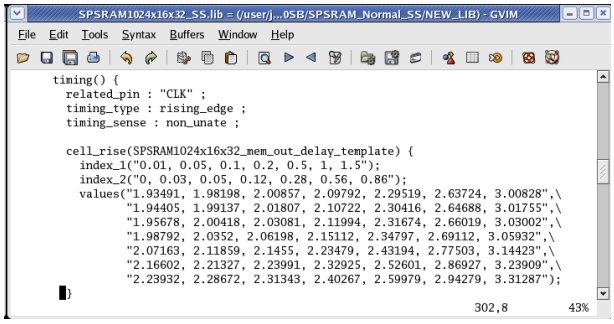

그림 13. Liberty 형태의 AMCM 타이밍 library
Fig. 13. AMCM timing library as a liberty format.

Ⅲ. 검증 결과

1. 제안된 방법론에 따른 특성 경향 비교 분석

본 논문에서 제안한 AMCM의 정확도를 검증하기 위하여 SPICE 시뮬레이션을 통하여 계산된 메모리 특성을 비교하였다. 그림 14~그림 19에서 보는 바와 같이 두 방법론 간의 차이는 엑세스 타이밍의 경우 약

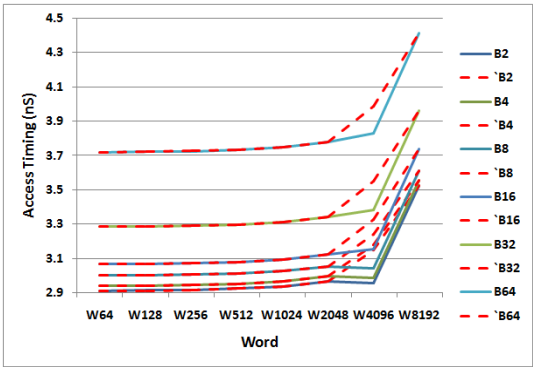

그림 14. Word 크기에 따른 엑세스 타이밍 경향성 비교
Fig. 14. Comparison of access timing for each word size.

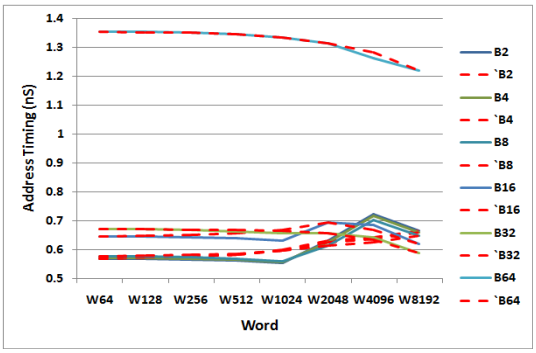

그림 15. Word 크기에 따른 어드레스 타이밍 경향성 비교
Fig. 15. Comparison of address timing for each word size.

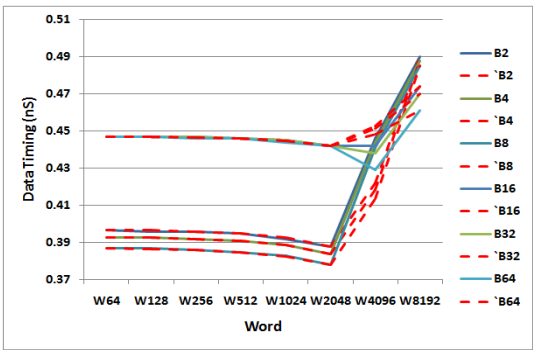

그림 16. Word 크기에 따른 데이터 타이밍 경향성 비교
Fig. 16. Comparison of data timing for each word size.

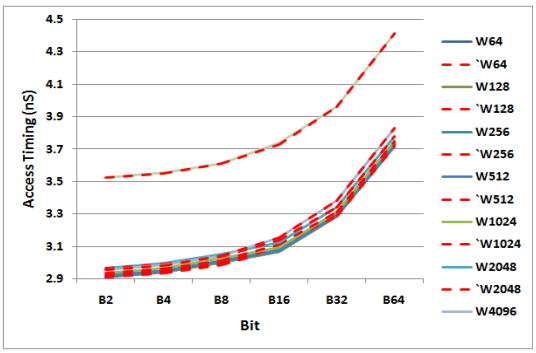

그림 17. Bit 크기에 따른 엑세스 타이밍 경향성 비교
Fig. 17. Comparison of access timing for each bit size.

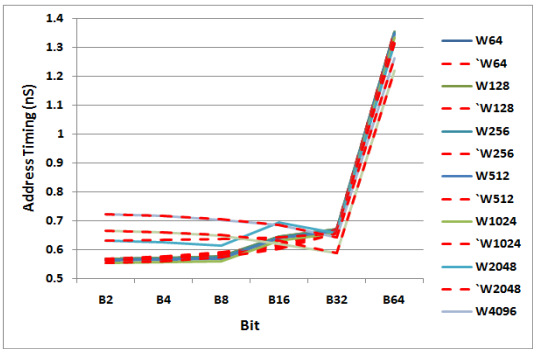

그림 18. Bit 크기에 따른 어드레스 타이밍 경향성 비교
Fig. 18. Comparison of address timing for each bit size.

-0.1%~+0.1%로 오차를 가지며, 어드레스 및 데이터 타이밍의 경우 약 -30pS~+ 20pS 오차를 가짐을 알 수 있다. 여기서 어드레스 및 데이터 타이밍에 대해서는 안정성을 기하기 위하여 상기 차이를 보상할 수 있을 만큼의 마진을 추가해준다. 본 논문에서는 실험한 결과에 따라 약 100pS의 마진을 추가하였으며, +120pS 더 크게 나왔다.

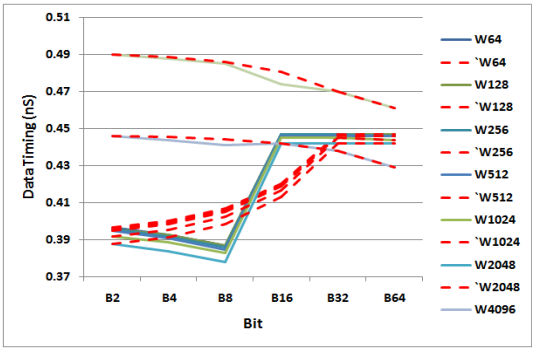

그림 19. Bit 크기에 따른 데이터 타이밍 경향성 비교
Fig. 19. Comparison of data timing for each bit size.

2. 제안된 방법론 유용성 검토

130nm 공정에 기반을 두고 개발된 메모리 컴파일러를 본 논문에서 제안한 AMCM을 사용하여 110nm 공정에 맞추어서 특성화하였다. 110nm 공정에 대하여 SPICE 시뮬레이션을 이용한 특성화한 결과와 본 논문에서 제안한 방법을 사용하여 특성화한 결과를 비교한 결과, 표 1에서 보는 바와 같이 엑세스 타이밍은 약 0.1~2% 내외, setup/hold 타이밍은 +120pS~+130ps, 전력 소모는 4% 내외의 차이를 가짐을 알 수 있었다. 같은 방법으로 180nm에서의 유용성을 확인한 결과 표 2와 같은 결과를 얻을 수 있었다.

표 1. 110nm 공정에 대한 결과 비교
Table 1. Compare result for 110nm process.

항목	SPICE	AMCM	Diff(%)	항목	SPICE	AMCM	Diff(%)
Access	1.839ns	1.8389ns	± 0.1 %	Cap.	0.116pF	0.123pF	± 1 %
Setup	302ps	419ps	+ 120 ps	Period	2.44ns	2.49ns	+ 2 %
Hold	212ps	341ps	+ 130 ps	Pulse Width	0.11ns	0.112ns	+ 2 %
Power	79.62pJ	82.8pJ	+ 4 %				

표 2. 180nm 공정에 대한 결과 비교
Table 2. Compare result for 180nm process.

항목	SPICE	AMCM	Diff(%)	항목	SPICE	AMCM	Diff(%)
Access	2.096ns	2.0955ns	± 0.1 %	Cap.	0.223pF	0.225pF	± 1 %
Setup	642ps	732ps	+ 110 ps	Period	3.65ns	3.72ns	+ 2 %
Hold	661ps	780ps	+ 120 ps	Pulse Width	0.16ns	0.163ns	+ 2 %
Power	100.73pJ	104.76pJ	+ 4 %				

3. 실리콘 제작을 통한 유용성 검증

그림 20은 AMCM을 사용하여 110nm 공정에 맞추어 특성화한 메모리 컴파일러를 이용하여 메모리 종류별, 크기별 다양한 메모리 20개를 생성하여 BIST를 적용한 테스트 칩에 대한 레이아웃이다.

그림 21은 그림 20의 칩을 실리콘 상에서 shmoo 테스트한 결과이다. 동작 전압은 1.0V~1.4V, 장비 제한 사항으로 동작 주파수는 25MHz~100MHz 까지 테스트하였으며, 결과를 보면 모두 정상적으로 동작했음을 알 수 있다.

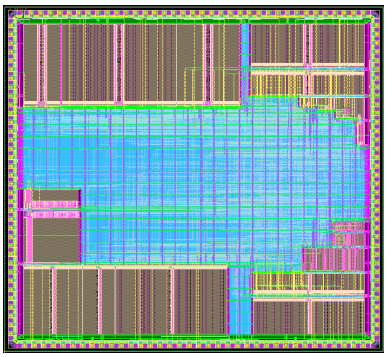

그림 20. 110nm 공정의 메모리 테스트 칩
Fig. 20. Memory test chip for 110nm process.

DVDD	1.8V	SP-SRAM (BIST Block) Y=1, X=4														
	nS	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
VDD																	
VDDc	1.000V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.040V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.080V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.120V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.160V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.200V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.240V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.280V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.320V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.360V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
VDDc	1.400V	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

그림 21. Shmoo 테스트 결과
Fig. 21. Shmoo test results.

Ⅳ. 결 론

본 논문은 주어진 공정에 맞추어서 메모리를 정확하고 빠르게 특성화할 수 있는 Advanced Memory Characterization Method를 제안하였다. 130nm 공정으로 제작된 메모리 컴파일러를 기반으로 본 논문에서 제안한 방법론을 사용하여 110nm 공정에 적합하도록 메모리를 특성화하였다. 특성화 결과와 SPICE 시뮬레이션을 통한 특성화 결과를 비교한 결과, 엑세스 타이

밍은 ±0.1%, setup/hold 타이밍은 +130ps로 오차 범위 내에서 유의성을 가짐을 확인하였다. 이를 검증하기 위하여 110nm 공정을 사용하여 테스트 칩을 설계하고 칩의 동작 및 수율을 확인한 결과. 실리콘 상에서 기능 검사의 평균 수율은 98.8%이었으며, 모든 메모리의 기능이 동작함을 확인하였다. 또한, 같은 방법으로 180nm 공정에 대하여 그 유용성을 확인한 결과, 기능 검사의 평균 수율은 98.3%이었으며, 모든 메모리 기능이 동작함을 확인 하였다. 본 논문에서 제안한 방법을 사용하여 SRAM 뿐만 아니라 register file 및 ROM(Read Only Memory) 등에 대해서도 실 제품에 확장 적용하고 있다.

REFERENCES

[1] Artisan Memory Compiler User Guide. 2009.
[2] Yen-Yu Chen, Shi-Yu Huang, and Yi-Chung Chang, “Rapid and Accurate Timing Modeling for SRAM Compiler,” IEEE International Workshop on Memory Technology, Design, and Testing. pp.73-76, Sept. 2009.
[3] Zhao-Yong Zhang, Chia-Cheng Chen * , and Jian-Bin Zheng, “A 90-nm CMOS Embedded Low Power SRAM Compiler,” IEEE Conference. ASICON ’09, pp.625-628, 0ct. 2009.
[4] Magma Design Automation, Inc.,SiliconSmart ACE User’s Guide. 2012.
[5] http://www.opensourceliberty.org/about_liberty.html
[6] Sharad Gupta, Parvinder Kumar Rana, “A 28㎚ 6T SRAM memory compiler with a variation tolerant replica circuit,” ISOCC Conference, pp.636-639, Nov. 2012.

저 자 소 개

신 우 철(정회원)
2003년 한국외국어대학교 전자공학과 학사 졸업.
2005년 한국외국어대학교 전자공학과 석사 졸업.
2007년 한국외국어대학교 전자공학과 박사 수료.
2007년～현재 동부하이텍(주) TE팀 책임연구원.
<주관심분야: SoC 설계, Low Power Design, Embedded Memory 설계, 설계자동화>

정 원 영(평생회원)
1988년 성균관대학교 물리학과 학사 졸업.
1996년 한양대학교 전자공학 석사 졸업.
2008년 숭실대학교 전자공학 박사 졸업.
1988년～1998년 LG반도체 선임연구원.
1998년～2000년 미국 Aspec/Ingenuus Cooperation 수석 연구원.
2000년～2003년 미국 Cadence Design System Sr. Engineering Manager.
2003년～2007년 미국 Nanno Solution, Inc. Exe. VP
2007년～현재 동부하이텍(주) 상무 Head of Technology Enabling.
<주관심분야: CAD & VLSI, DFM/DFY, TCAD/ESD Simulation & SPICE Modeling>

송 혜 경(정회원)
1996년 경기대학교 전자계산학과 학사 졸업.
1996년～2007년 ㈜삼성전자 LSI 사업부.
2007년～현재 동부하이텍(주) TE팀 수석연구원.
<주관심분야 : SoC설계, embedded Memory 설계, 설계자동화>

조 경 순(평생회원)-교신저자
1982년 2월 서울대학교 전자공학과 학사 졸업.
1984년 2월 서울대학교 전자공학과 석사 졸업.
1988년 12월 미국 Carnegie Mellon University 전기 및 컴퓨터 공학과 박사 졸업.
1988년 11월～1994년 8월 삼성전자(주) 반도체 총괄 선임, 수석 연구원.
1994년 8월～현재 한국외국어대학교 전자공학과 조교수, 부교수, 정교수.
<주관심분야: SoC 설계>