

ASIC 메모리 콤파일러 라이브러리 특성 추출기법

A Characterization Method for ASIC Memory Compiler Library

정 승민 (정보통신과)

Seung-Min Jung (Dept. of Information and Communication)

Key Words : Memory Compiler(메모리 모듈생성기), Characterization(특성추출), Delay(지연시간), ASIC(주문형 반도체 집적회로)

ABSTRACT: In general, it is impossible to characterize all configuration of memory compiler which can be generated. So, we need to characterize many configurations as possible for more accurate timing of memory compiler library. This paper suggests the new method for more accurate and speedy results of memory compiler library characterization. That includes advanced delay model and methodology of memory compiler library characterization.

1. 서론

아식칩(ASIC Chip) 설계에 있어서 Embedded DRAM technology 가 적용되고 있는 시점¹⁾ 이긴하나 공정개발이 더디고 아직 신뢰성이 부족하므로 고성능 Embedded SRAM 콤파일러의 필요성은 여전히 중요하다. 다양한 사이즈의 메모리를 칩 설계시 적용할 수 있다는 것은 아식 칩 설계자로서 가져야 할 당연한 설계환경이며 이에 아식 메모리 콤파일러를 개발하는 라이브러리 개발자는 이를 충족시킬 수 있는 고성능, 고신뢰성의 메모리 콤파일러를 구축 해야한다. 하지만 빠르게 진보하고 있는 공정기술의 발달로 인하여 메모리 콤파일러의 회로기술개발이 공정개발을 따라가기에 급한 실정에 있다. 이는 technology 향상에 따른 능동적인 특성추출(이하, characterization 으로 표기) 환경을 구축해 놓음으로써 훨씬 쉽게 대처할 수 있다. 메모리 콤파일러의 characterization 은 일반 단순 logic 라이브러리와는 달리 정확성을 위한 characterization flow 가 매우 복잡하고 시간 또한 많이 소요된다. 메모리 콤파일러 characterization 의 가장 큰 과제중의 하나는 generation 할 수 있는 모든 configuration 에 대한 characterization 데이터베이스를 구축하기 힘들다는 점이다. 따라서 전체 라이브러리 characterization 의 정확성을 위해서는 가능한 많은 configuration 에 대한 characterization 을 실시하는 것이 중요하고 이는 결국 전체 라이브러리의 characterization 에 소요되는 시간의 문제로 귀결된다.

본 논문에서는 메모리 콤파일러의 개발단계 중 특성 추출과정에 있어서 좀 더 진보된 방법을 제안하고자 한다. 메모리의 전달 지연시간(propagation delay) 추출을 위한 방법을 제안하고 실제 메모리 instance 를 이용하여 기존 방법으로서의 HSPICE simulation 결과와 새로운 방법에 의한 결과를 비교함으로써 해서, 정확성의 검증과 함께 실제 응용범위에 대해서도 언급하였다. 이는 characterization 의 정확도를 향상시키면서 아울러 characterization 에 소요되는 시간 또한 줄이는 방법으로 향후 technology upgrade 에 따른 능동적인 메모리 콤파일러 개발을 위한 characterization system 환경을 구축할 수 있는 진보된 방법이 될 것이다.

2. 지연시간(delay) 모델

일반적으로 셀(이하, cell 로 표기함) 의 전달 지연시간(propagation delay) 은 다음 식과 같이 intrinsic 지연시간(이하, delay 로 표기함)와 routing 요소인 load capacitance CL 에 의한 delay 로 나뉘어진다. [2][3]

$$\text{tpd}(\text{TS}, \text{CL}) = t_i + t_s \quad (1)$$

여기서 $\text{tpd}(\text{TS}, \text{CL})$ 는 input slope TS 와 load CL 에 의한 cell 의 전달 지연시간(propagation delay) 이며, t_i 는 cell 의 자체고유 delay 이고 TS, CL 에 따라 변할 수 있다. 한편, t_s 는 load CL 에 의한 순수 routing delay 로서 cell delay 가 아닌 CL 만의 delay 성분 이다. 일반적으로 메모리 콤파일러를 포함한 cell 라이브러리 characterization 은 discrete 한 TS, CL 의 각 point 에 대하여 그림 1 과 같이 메모리 cell 의 critical path netlist 에 대해 HSPICE simulation 을 하여 TS, CL 의 delay table 을 구하는 flow 를 적용하고 있다. 이런 방식으로 가능한 많은 메모리 configuration 에 대한 simulation 을 실시한다. Delay calculator 의 interpolation 과 extrapolation 의 정확성을 위해서는 simulation 된 configuration 갯수가 매우 중요하며 TS, CL 을 어떠한 값, 몇 개의 point 로 설정했느냐에 따라 결정된다.

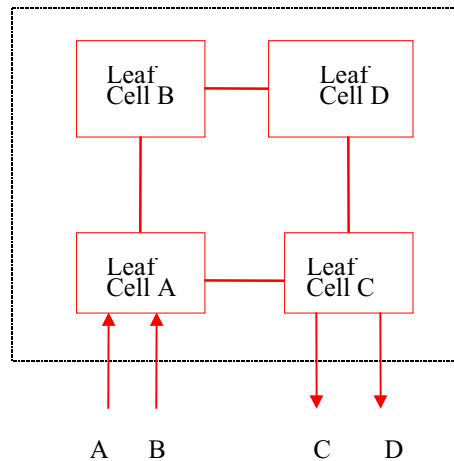


그림 1 메모리 콤파일러 특성추출을 위한 일반적인 critical path 구조

본 논문에서는 식 (1) 의 지연시간식(delay equation) 을 좀더 세분화하여 TS, CL 의 변화에 constant 한 delay 성분과 그렇지 않은 성분을 구분한다. [2][4][5] 세분화된 cell 의 새로운 전달 지연시간(propagation delay) 은 다음식과 같다.

$$\begin{aligned} \text{tpd}(\text{TS}, \text{CL}) &= t_i + t_s \\ &= t_{i_load}(\text{TS}) + t_{i0}(\text{constant}) + t_{i_drive}(\text{CL}) + t_s(\text{CL}) \end{aligned} \quad (2)$$

여기서 t_{i0} 는 최소 intrinsic delay 성분으로 상수값이며, t_{i_load} 는 intrinsic delay 의 TS dependency 성분이고, t_{i_drive} 는 intrinsic delay 의 CL dependency 성분이며, t_s 는 cell delay 가아

닌 routed CL 만의 delay 성분이다.

이렇게 하면 cell delay 각 성분을 구하는데 있어 critical path 또한 구분 되어져야 한다. critical path 구분에 따른 simulation mode 를 본 논문에서는 다음과 같이 새롭게 정의한다.

- **Intrinsic 지연시간을 구하기위한 A mode simulation**
- **TS, CL 변화에의한 지연시간을 위한 B mode simulation**

각 mode 에서 사용하는 critical path 는, A mode 의 경우 그림 2 (a) 와 같이 critical path schematic 전체를 사용하는 기본 simulation 이다. 이 simulation 에서는 최소 TS 와 최소 CL 에 대한 simulation 을 통하여 delay 성분중 t_{i0} 를 구하게 된다. 반면 B mode simulation 에서는 normal critical path 에서 사용하는 load cell, bit cell 및 IO pin 을 갖고 있지 않은 내부 leaf cell 을 제외한 peri-leaf cell 만으로 simulation netlist 를 다시 구성한다. TS, CL 에 대한 전체 cell delay dependency 는 IO pin 을 갖고 있는 peri-leaf cell 의 simulation 만으로도 충분하다.^{[6][7][8]} B mode simulation 을 통하여 다음의 지연시간식(delay equation) 계수 성분을 얻게된다.

- **Input pin 의 TS dependency equation 의 계수**
- **Output pin 의 CL dependency equation 의 계수**
- **load CL 을 drive 할 power 모델을 위한 계수**

B mode simulation 의 목적은 정확성을 최대한 유지하면서 characterization 소요시간 을 줄이는데 있다. 본 논문의 특성 추출 방법에 따르면 일반적인 메모리 아키텍처에 있어서 그림 2 (b) 와 같은 peri-block 은 generation 되는 메모리 콤파일러 configuration 에 관계없이 동일한 leaf cell 을 사용하게 되며 따라서 각각의 configuration 에 대하여 TS, CL 의 변화에 대해 반복적으로 simulation 할 필요가 없는 것이다.

정리하면 하나의 configuration 에 대하여 t_{i0} 성분을 위한 최소 TS 와 최소 CL 에 대한 A mode simulation 은 반드시 실시 하며 동일한 peri-block 의 구조인 메모리 콤파일러인 경우 라이브러리 전체를 통틀어 TS, CL 의 각 point 에 대한 t_{i_load} (TS), t_{i_drive} (CL) 성분 추출을 위한 B mode simulation 은 단 1 회만으로 끝나게 된다.

최종 characterization 값은 A + B mode simulation 결과이며 각 configuration 에 대한 반복적인 simulation 은 A mode 만 수행되면 된다. 대부분의 메모리 콤파일러의 아키텍처를 보면 이러한 delay 추출방식에 대하여 쉽게 이해 할 수 있는데, 이들의 critical path schematic 은 각 block 별로 복잡하고도 많은 수의 TR 을 갖고 있으며 각 leaf cell 또한 LPE 추출 netlist 를 갖고 있어 TS, CL 에따른 막대한 simulation 소요시간을 필요로 한다. 따라서 1 개의 TS/CL 조건에대해서만 전체 critical path 를 사용하고(A mode simulation) TS/CL dependency 는 peri-leaf cell 만으로 구성된 peri-netlist 만을 재구성하여 B mode simulation 을 함으로써 simulation 소요시간을 획기적으로 줄일 수 있다. 물론 최종 cell delay 는 A + B mode simulation 결과이다.

3. delay 모델을 위한 UDB(User Data Base) 의 정의

본 논문에서는 characterization 을 control 하고 결과를 저장하는 기본 운용 data base 를 별도로

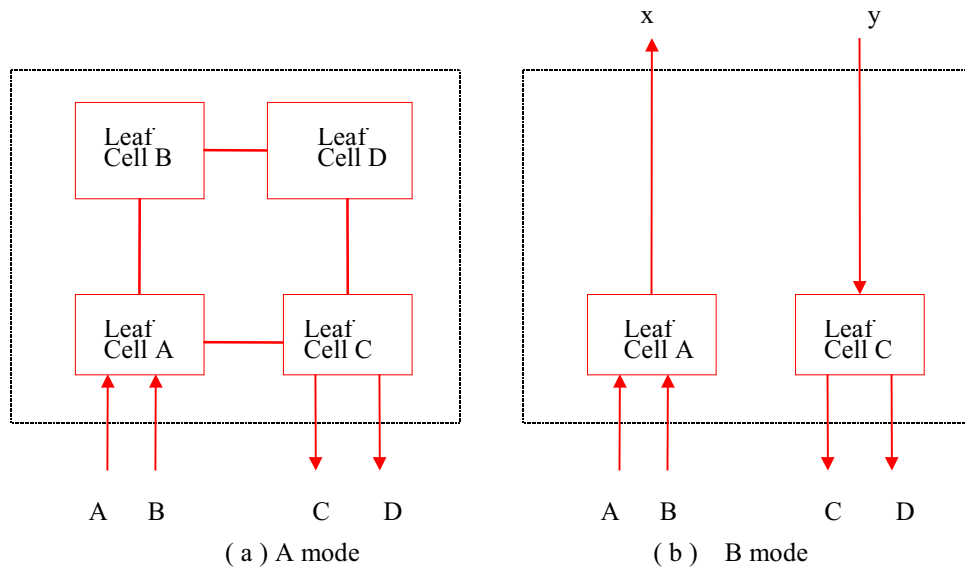


그림 2 mode 별 simulation critical path 회로도

정의하는데, UDB(User Data Base) 로 정의한다. 이 UDB 는 각 delay 를 simulation mode 별로 HSPICE input file 을 만들고 simulation 결과를 저장한다. 본 논문에서는 UDB 를 운용하기 위한 자체 프로그램을 제작하였으나 이를 자세히 언급하는 것은 논문의 방향과 일치하지 않으므로 본 논문의 이해를 위한 최소한의 field 만을 언급하기로 한다. 본 논문에서 정의된 UDB 를 구성하는 기본 field 와 delay 성분 , simulation mode 와의 매칭관계는 다음 표 1 과 같다.

표 1 정의된 UDB field 와 delay 성분, simulation mode 와의 관계

Parameter Field in 최종 UDB	Delay 성분	Simulation Mode	Finally Used Data base 의 형태
P	ti	A+B mode	Table format
T	ts	B mode	Table format
V	ti0	A mode	Constant
S	timing hazard	A+B mode	Table format
Tc	ti_load(TS)	B mode	Equation format
Cc	ti_drive(CL)	B mode	Equation format

서두에서도 언급한 바와 같이 characterization 결과는 크게 두 종류로 나뉜다. 첫째는, A mode simulation 결과인 ti0 에 B mode simulation 의 TS, CL 의 dependency 에 따른 실제 characterization 결과를 더한 table 데이터이다. 둘째는, B mode simulation 으로부터 얻은 메모리 Input/Output Pin 의 TS, CL dependency equation 의 계수 (Tc, Cc) 와 load CL 의 drive 를 위한 Output Pin 의 DC parameter 를 포함하는 데이터 인데 이는 table 결과로부터 curve fitting 을 통하여 얻어지는 계수 데이터 이다. 그 밖에 Input/Output capacitance , Input Pin 의 logic threshold , Output maximum capacitance 등의 parameter 를 포함한다. 이상은 본 논문에서 equation format data 로 정의한다. Delay 계산에 사용되는 data 는 table 형태와 equation 모두가 가능하며 ASIC 설계자가 어떠한 환경에서도 사용할 수 있도록 데이터 베이스를 구축해 놓아야한다. 본 논문에서는 TS, CL 이 각각 3 개의 값으로 정의되는 3 X 3 lookup table 모델을 적용하였다. 물론 어떠한 N X N 모델을 적용해도 상관은 없다.

3-1 . 전달지연시간(tplh/tphl) 계산

UDB 의 table delay format 은 위에 언급한 바와 같이 P field 값 과 T field 값 이다. 각 TS point 에 대한 P field 값 성분은 다음 3 가지 값으로 계산된다.

$$\begin{aligned}
 \bullet \quad p_0 &= V + (P_0 - P_0) - C_0 & (3) \\
 p_1 &= V + (P_1 - P_0) - C_0 \\
 p_2 &= V + (P_2 - P_0) - C_0,
 \end{aligned}$$

이 경우 t_i 성분인 V field 값 에는 이미 C_0 가 포함되어 있으므로 순수한 t_i 에서는 이 값을 제외시켜야 한다. 한편 ts 의 경우 T field 값을 보면,

$$\begin{aligned}
 \bullet \quad T_0 &= C_0(AVE) & (4) \\
 T_1 &= C_1(AVE) \\
 T_2 &= C_2(AVE)
 \end{aligned}$$

로 표현되며 이 값은 그대로 적용될 수 있다. T field 값의 경우 실제 B mode simulation 에서 TS 변화에 대한 T 값의 변화도 고려하여 이들의 미묘한 변화량을 구하여 평균을 취한 C(AVE) 를 구하게 된다. 그림 3 과 그림 4 (a),(b),(c) 를 참고하면 언급한 각 delay 성분에 대한 이해가 빨라진다. 최종 UDB 에서 $p_{0,1,2}$ 의 각 값은 그림 3 과 같이 “P” field 로, $T_{0,1,2}$ 각 값은 “T” field 로 저장된다. 실제 tplh/tphl 을 구하기 위해서는 UDB file 의 data 인 P field 값과 T field 값을 이용하여 다음과 같이 구할 수 있다.

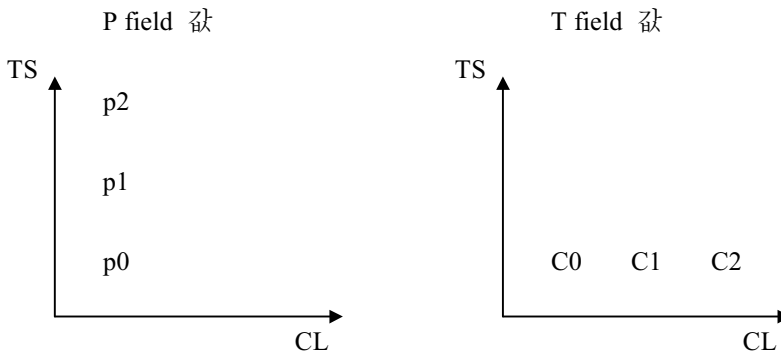
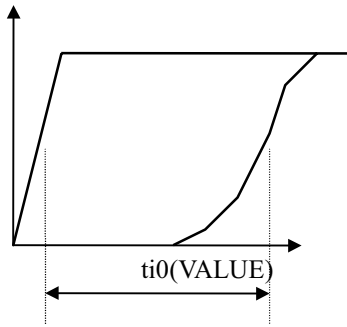


그림 3 P field 값 과 T field 값 의 관계

최종 3X3 지연시간 table 은 다음식과 같다.

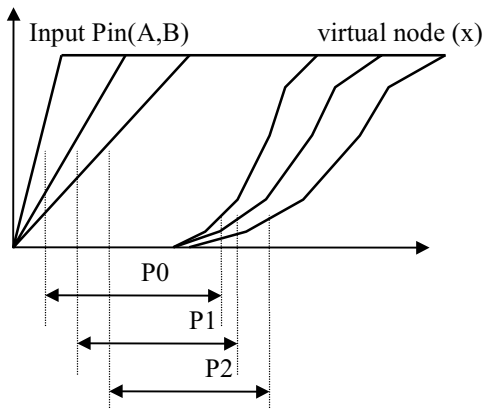
$$\begin{array}{ccc}
 \text{tplh/tphl} = & & \text{CL} & & (5) \\
 & p_0+C_0 & p_0+C_1 & p_0+C_2 & \\
 & p_1+C_0 & p_1+C_1 & p_1+C_2 & \text{TS} \\
 & p_2+C_0 & p_2+C_1 & p_2+C_2 &
 \end{array}$$

tr/ta의 경우 T field 값을 이용하여 표현할 수 있을 것이다.



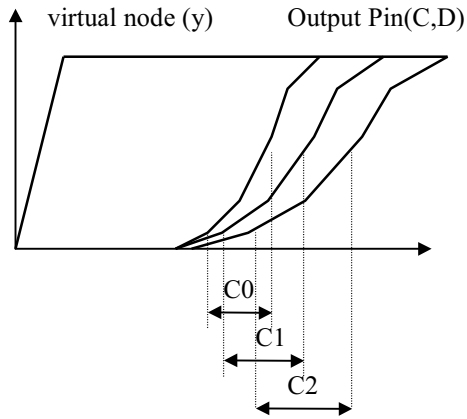
<A mode simulation>
 @최소 TS, 최소 CL
 ti0 => 최종 UDB 의 “V” field

(a)



<B mode simulation @input pin>
 최소 CL @virtual output node(그림 2(b) X)

(b)



<B mode simulation @output pin>
 최소 TS @internal input node(그림 2(b) Y)

(c)

그림 4 simulation mode 별, delay 성분별 곡선

3-2. Timing Hazard 계산

B mode simulation 은 앞서 언급한 바와 같이 Input/Output Pin 에 대한 TS/CL 의 변화에 대한 dependency 를 추출한다. 좀더 정확히 말하면 cell 의 intrinsic delay 성분인 ti 의 변화량을 추출한다. B mode simulation 의 그림 5 로 부터 각 input/output pin 에 대하여 TS/CL 에 대한 지연시간식(delay equation) 을 만들 수 있다.

$$ti(TS,CL) = a + b*TS + c*TS^2 + CL*(d + e*TS + f*TS^2) + CL^2*(g + h*TS + i*TS^2) \quad (6)$$

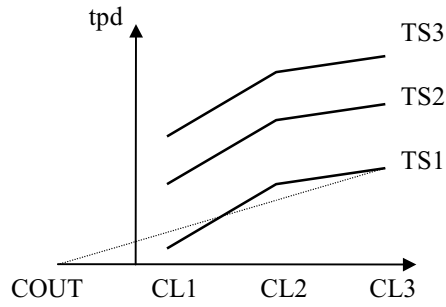


그림 5 지연시간식(delay equation) 추출 관계

그림 5 는 그림 4 의 delay 곡선을 재구성한 것이고 curve fitting 을 통하여 식 (6)과 같이 $ti(TS,CL)$ 에 대한 지연시간식(delay equation) 계수 (a, .., i) 를 구하게 된다. 각 계수 는 Input pin 의 경우 $ti_load(TS) = ti(TS,CL) \{ @최소 CL \} = j + k*TS$ 로 , Output pin 의 경우 $ti_drive(CL) = ti(TS,CL) \{ @최소 TS \} = 1 + m*CL$ 로 근사화 된다. 메모리의 경우 2차 성분은 전체 delay 에 비하여 미미하므로 primitive cell 과는 달리 제외되어 1차 방정식으로 근사화 된다. 이때 얻어지는 계수, j, k 는 TC filed 로, l, m 은 CC field 로 저장된다.

setup/hold 및 pulse width 를 추출하기위해서 B mode simulation 으로부터 얻은 $ti_load(TS)$ equation 을 이용한다. 예를 들어 WEB(write enable low signal pin) to DI(data input bus pin) setup time 의 경우 각 pin 의 지연시간식(delay equation) 을 $ti_load_WEB(TS) = c1 + c2*TS1$, $ti_load_DI(TS) = c3 + c4*TS2$ 라고 하면 TS 변화 에따른 setup time 의 dependency 는 $ti_load_WEB(TS) - ti_load_DI(TS) = V = c0 + c2*TS1 + c4*TS2$ 로 표현된다.

setup/hold 및 pulse width 는 알고 있는 바와 같이 control pins 과 controled pins 의 TS dependency timing 이므로 지연시간(delay) 과는 다른 table index 를 사용하게 된다. 그림 6 의 그래프를 통하여 위 관계식을 쉽게 이해할 수 있다. 물론 위 관계식과 그래프는 TS dependency 만을 표현한 것이며 A mode simulation 으로부터 구해진 V field 값에 TS dependency 를 포함한 table 형태의 결과는 최종 UDB 에서 S field 에 저장된다. 각 TS1, TS2 에 대한 timing hazard table 은 언급한 다음의 식 $V = c0 + c2*TS1 + c4*TS2$ 로부터 각 TS 에따른 결과를 정리하면 다음과 같다.

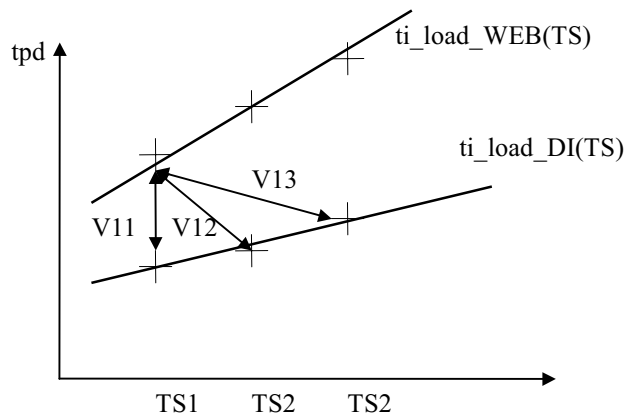


그림 6 Timing Hazard 의 계산을 위한 관계

	TS1	TS2	TS3
TS1	V	$V+(V12-V11)$	$V+(V13-V11)$
TS2	$V+(V21-V11)$	$V+(V22-V11)$	$V+(V23-V11)$
TS3	$V+(V23-V11)$	$V+(V32-V11)$	$V+(V33-V11)$

여기서 $V = c0 + c2*TS1 + c4*TS2$ (@최소 TS1 & 최소 TS2) 이며 최종 UDB 의 S filed 는 table format 의 output 으로 사용된다.

4. Circuit Partitioning

B mode simulation 을 위한 netlist 는 critical path 에서 IO 를 갖는 peri-block 으로 구성되며 configuration dependency 관계가 있는 load cell 은 제외된다. input pin 에 대해서는 internal output node (그림 2 (b) node X) 를 지정해 주어야하고 output pin 에 대해서는 internal input node (그림 2 b node Y) 를 지정해 주어야 한다. Simulation node point 를 설정하는데는 어느정도의 기준이 있다.

- normally 3rd logic stage
- internal node 에는 configuration dependency 가 없도록 한다.(그림 7 (a))
- parallel path 가 있으면 정확성이 떨어진다.(그림 7 (b))
- 동일 pin 에 2 개이상의 logic threshold (Vth) 가 없도록 한다.(그림 7 (c))
- ADD 와 DI 와 같이 bus input 인 경우 동일한 input cap 을 위해 같은 트랜지스터 크기이어야 한다.

그림 7 (a) 의 경우 TC 가 load cell 에 dependency 하므로 configuration 에 따른 정확성을 확보할 수 없다. TC 의 경우 최소 CL 을 적용하도록 되었다는 점을 생각하면 문제의 이유를 쉽게 이해할 수 있다. 그림 7 (b) 의 경우 동일한 pin 에 3 개의 TC 가 존재하지만 본 논문에서는 이중 1 개만을 데이터화 할수있도록 하였으므로 나머지 두개의 path 를 갖는 delay 나 timing hazard 는 accuracy 를 보장할 수 없다. 이 경우 일단 TS 변화에 따른 dependency 가 가장 큰 path 를 택한다. 그림 7 (c) 의 경우 동일 node 에 2 개의 logic threshold 가 존재하는 경우로 이 경우 본 논문에서는 평균을 취하기로 하였다.

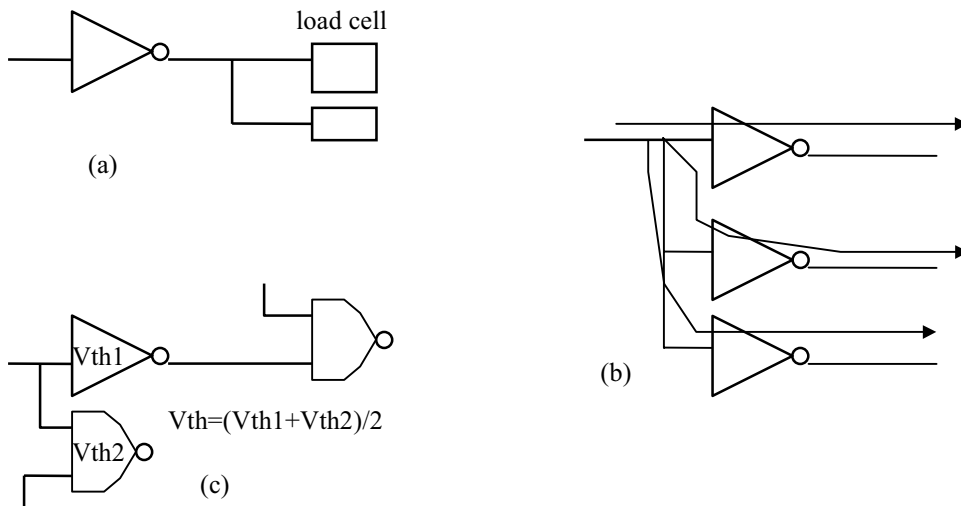


그림 7 B mode simulation circuit (not recommended)

참고자료 1 에 본 논문에 적용된 메모리 instance 의 B mode simulation netlist 를 나타내었다. 본 논문의 메모리 아키텍처는 일반적인 메모리 콤파일러의 모든 제약 조건에 해당되게 함으로써 B mode simulation 정확도가 떨어질 것으로 예상되었다. 결국 본 논문의 지연시간 (delay) 모델 을 이용한 characterization 에서 정확한 B mode simulation 이 되기 위해서는 아키텍처의 제약조건이 있으며 정확한 characterization 을 위해서는 설계될 메모리 아키텍처는 이러한 모든 조건을 만족해야 한다. 향후 메모리 characterization 에 있어서 개선되어야 할 사항이다.

5. Characterization 최종 결과 및 비교분석

본 논문에서는 제안된 characterization 방법의 결과를 비교하기 위한 샘플 data 로서 전형적인 0.35 um two-port SRAM 1024x4 module 을 선정하였다. 결과 분석은 표 2 와 같다.

thzp/thzn 을 포함한 delay parameters 의 경우 taoer/taoef 경우를 제외한 나머지 parameters 에 있어서는 공통적으로 rising 과 falling 어느 한쪽은 3 %이내의 정확한 결과를 보이지만 나머지 한쪽은 10-20 % 대의 높은 평균오차 범위를 보인다. 이와 같은 결과를 보인 이유는 본 논문에서 A mode simulation 의 결과가 $\max(V_{\text{rising}}, V_{\text{falling}})$ 를 취하게 내부적으로 setting 이 된 상태에서 결과이므로 실제 HSPICE 결과와는 rising/falling 값중 한쪽만 일치하는 현상을 나타내었다. 물론 이는 내부 프로그램을 수정하여 rising/falling parameter 를 별도로 선언해주고 이에 따라 UDB template 에도 별도로 선언을 해주면 모든 값이 정확하게 추출될 것이다. 향후 추출의 결과 모든 값이 3 % 이내의 결과를 얻을 것으로 예상된다. 한편 taoer/taoef 의 경우 오차가 큰 편인데 두 가지 요인을 생각할 수 있다. 하나는 B mode simulation schematic 에서 oeb pin 의 virtual output node OE 가 configuration 에 따른 load cell 을 갖고 있지만 실제 B mode simulation 에서는 이러한 영향을 배제하고 있다. 그러나 기준자료(reference data) 추출시에는 이를 고려하고 있다. 메모리의 architecture 제한 조건에 의한 원인으로 볼 수 있다. 두 번째 요인은 taoe characterization 값 자체가 1 ns 이내로 같은 오차에 대하여 다른 access time 보다 오차율이 클 수 밖에 없고 characterization 에 대한 오차가가능성도 그만큼 크다고 할 수 있다.

Timing hazard 경우 본 논문에서는 값에 marginal 한 요소가 부가되도록 하였다. 이는 characterization 방법이 scanning 이 아닌 내부 몇 개의 node simulation time (t(A),t(B),..) 에 대한 계산값 이므로, 이는 계산을 했다는 의미자체에 부정확성의 요소가 내포되어 있는 것이다. timing value 자체가 작은 상태에서 marginal 한 계산을 거치게 되어 기준자료(reference data) 와는 오차율이 커지게 된다. 물론 기준자료(reference data) 도 marginal 한 계산결과이다. 또 하나의 오차 원인으로서는 앞장의 그림 5 (b) 에서도 언급한 바와 같이 web , wax pin 의 rising/falling 의 TC 을 추출하는 circuit 의 path 가 동일하지 않다는 점인데 예를 들어 web 의 rise_Tc 은 node we4 를, fall_Tc 은 we2 까지의 delay simulation 으로 추출하게 되어있으므로 fall_Tc 와 같은 경우 address setup time 의 계산에 있어서 정확하지 않은 값을 별수없이 적용해야하는 아키텍처 problem 을 갖고 있는 것이다. din_setup time 의 경우는 반대로 rise_Tc 이 적용되어 정확성을 저하시킨다. 이러한 이유는 이미 언급한 것처럼 하나의 input pin 에 여러가지의 path logic 이 존재하지만 UDB 는 이중 one path 만을 적용할 수밖에 없는 data base 구조를 갖도록 하였기 때문이다. 또 하나의 공통적인 원인의 하나로 본 논문에서는 actual input logic threshold 를 적용하였다는 점이다. HSPICE 의 경우 vdd/2 로 일괄 measure 하게 되어있으나, 자체 B mode simulation 에서 나온 VTH 를 적용함으로써 해서 timing hazard 와 같은 작은 값의 characterization

표 2 0.35 um two-port SRAM 1024x4 결과 비교(Temp 25 , Typical Process)

(음영 : 기존 방식의 HSPICE simulation 결과 , 굵은 수: new method 에 의한 simulation 결과)

		delay parameter (ns)									
TS(ns)	CL(pF)	tacr	tacf	tarer	tafef	Taoer	taoef	twdor	twdof	tdidor	tdidof
0.1	0.0	3.707	4.495	3.793	4.612	0.433	0.303	4.636	5.457	3.793	4.991
new method 결과		4.495	4.495	4.622	4.622	0.433	0.433	5.404	5.404	5.021	5.021
0.1	0.25	3.768	4.462	3.839	4.666	0.513	0.378	4.697	5.511	3.856	5.048
new method 결과		4.520	4.526	4.647	4.653	0.459	0.465	5.429	5.435	5.046	5.052
0.1	0.5	3.821	4.513	3.892	4.717	0.572	0.436	4.750	5.558	3.908	5.095
new method 결과		4.550	4.542	4.677	4.669	0.489	0.481	5.459	5.451	5.076	5.068
0.5	0.0	3.683	4.549	3.844	4.712	0.513	0.370	4.708	5.492	3.846	4.978
new method 결과		4.520	4.520	4.641	4.641	0.452	0.452	5.435	5.435	5.063	5.063
0.5	0.25	3.741	4.610	3.900	4.768	0.595	0.452	4.764	5.553	3.901	5.045
new method 결과		4.546	4.552	4.667	4.673	0.478	0.484	5.461	5.466	5.089	5.094
0.5	0.5	3.794	4.667	3.954	4.815	0.654	0.510	4.821	5.610	3.955	5.089
new method 결과		4.576	4.568	4.697	4.689	0.508	0.500	5.490	5.483	5.119	5.111
2.0	0.0	3.629	4.715	4.037	4.868	0.803	0.631	4.810	5.642	4.033	5.374
new method 결과		4.556	4.556	4.693	4.693	0.463	0.463	5.515	5.515	5.128	5.128
2.0	0.25	3.675	4.782	4.099	4.925	0.870	0.708	4.866	5.706	4.083	5.429
new method 결과		4.582	4.588	4.719	4.725	0.489	0.494	5.541	5.546	5.154	5.160
2.0	0.5	3.727	4.834	4.152	4.973	0.932	0.768	4.920	5.756	4.146	5.453
new method 결과		4.612	4.604	4.749	4.741	0.518	0.511	5.570	5.562	5.184	5.176
평균오차(%)		21.6	1.9	18.2	2.1	22.5	21.6	13.3	1.8	28	1.6

결과에 대하여 높은 오차 영향을 미쳤을 것으로 생각된다. 물론 최종 UDB 를 merge 할 때 정의된 vdd/2 로 적용하는 것도 가능하다. tWDS 의 경우 전반적으로 정확하지만 TS 가 2.0 N 인 경우 오차가 매우 커지는 현상을 볼 수 있는데 앞서 언급한 것처럼 logic threshold 를 적용한다면 정확도가 높아질 것으로 예상된다. tWP, tWPC 의 경우는 문제가 좀더 심각한데, 값의 경향이 정 반대이다. 가능한 분석으로, B mode simulation 용 UDB template 의 tWP, tWPC 에 대한 timing measure 정의에 오류가 있는 것으로 보인다. 이는, HSPICE simulation 때의 characterization point 와 본 논문에서 적용하는 point 가 다르기 때문인 것으로 여겨지며 추후 적용시에 추가로 검증이 요구되거나 충분히 해결할 수 있는 문제이다.

6. 결론

본 논문에서는 기존 lumped 모델 형태의 메모리 콤파일러 characterization 방식과는 다른 진보된 delay 모델과 이를 기반으로 하는 새로운 방식의 메모리 characterization 방법을 제안하였다.

표 2. (계속)

Condition		Timing Hazard	
TS1(ns) (WEB)	TS2(ns)	twas	twds
0.1	0.1	0.415	0.767
New method 결과		0.415	0.771
0.1	0.5	0.444	0.834
New method 결과		0.472	0.817
0.1	2.0	0.495	1.212
New method 결과		0.575	0.992
0.5	0.1	0.352	0.743
New method 결과		0.387	0.737
0.5	0.5	0.382	0.810
New method 결과		0.421	0.784
0.5	2.0	0.433	1.188
New method 결과		0.315	0.958
2.0	0.1	0.193	0.751
new method 결과		0.349	0.567
2.0	0.5	0.225	0.819
new method 결과		0.456	0.613
2.0	2.0	0.275	1.197
new method 결과		0.415	0.788
평균오차(%)		33	12.8

Condition		Timing Hazard		Delay	
TS1(ns)	TS2(ns)	twp	twpc	thzp	thzn
0.1	0.1	1.986	6.012	0.273	0.306
Parquet 결과		1.923	5.200	0.273	0.273
0.5	0.5	2.071	6.088	0.306	0.391
Parquet 결과		1.889	5.172	0.293	0.293
2.0	2.0	2.185	6.267	0.306	0.346
Parquet 결과		1.718	5.100	0.294	0.294
평균오차(%)		11	15	2.6	17

tacr/f : ADD change to DOUT rising / falling access time
 tarer/f : REB falling to DOUT rising/falling access time
 taoer/f : OEB falling to DOUT rising/falling access time
 twdor/f : WEB falling to DOUT rising/falling access time when contention mode
 tddor/f : DI rising/falling to DOUT rising/falling when write mode

twas : WEB to ADD setup time
 twds : WEB to DI setup time
 thzn/p : OEB rising to DOUT Hi-z time
 twp : WEB minimum pulse width low
 twpc : WEB minimum pulse width high

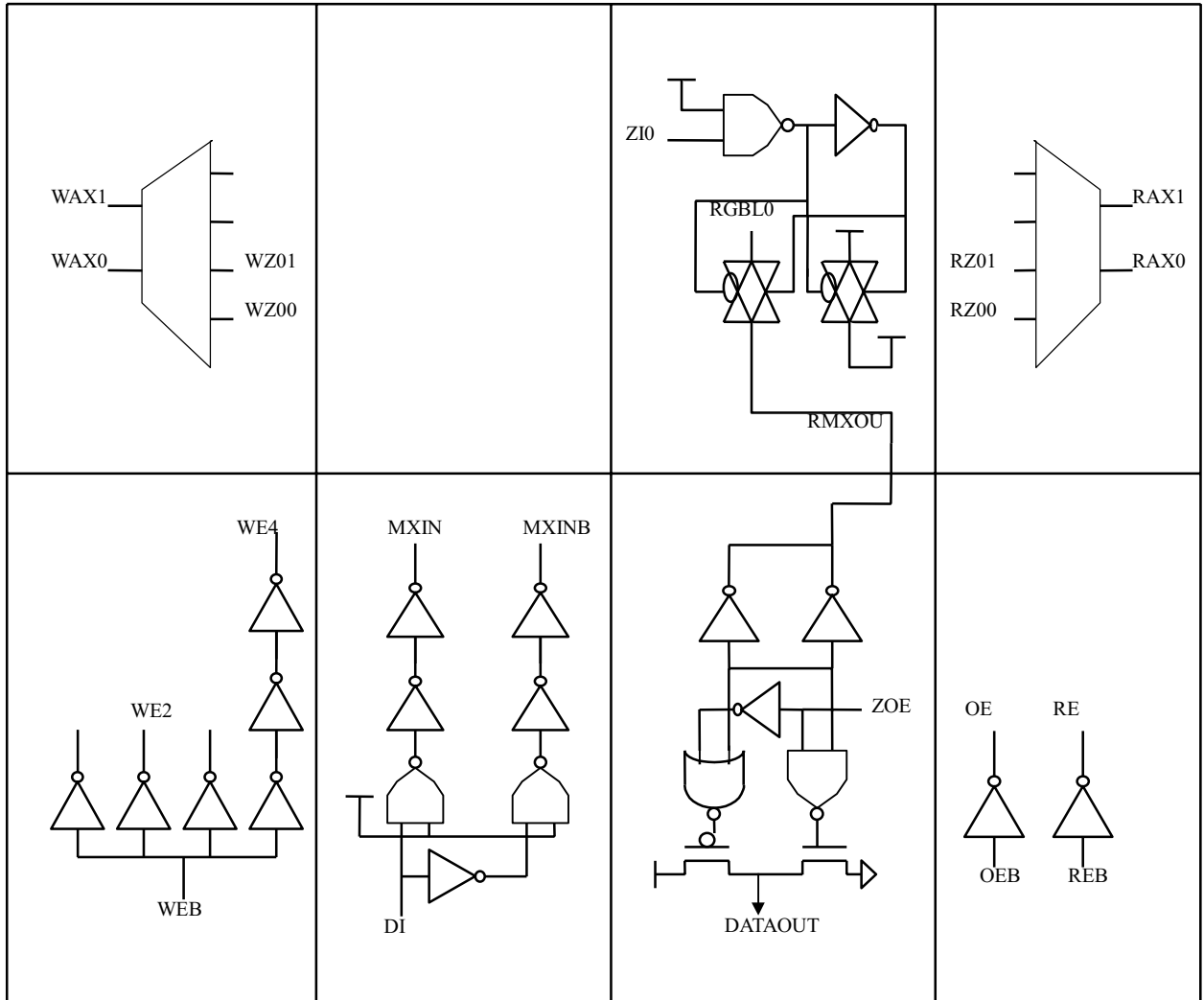
delay 모델 에 있어서는 TS, CL 의 변화에 dependant 한 성분과 constant 한 성분으로 구분하였고 이에 따른 A simulation mode , B mode simulation mode 를 정의 하였고 각 mode 에 따른 메모리 critical path 를 별도로 정의 하였다. 특히 B mode simulation 을 위한 critical path circuit partitioning 기법을 언급하였다. 정의된 delay 모델의 각 성분을 정의하기 위한 자체 data base format 인 UDB 를 정의 하였으며 이는 전체 simulation 을 control 하고 characterization 결과를 저장하며 자체 프로그램에 의해 운용된다.

메모리 아키텍처 구조상 B mode simulation 은 전체 메모리 라이브러리 characterization flow 에 있어서 단 1 회만 수행되면 되므로 기존 메모리 characterization 방법에 비하여 소요시간을 혁신적으로 줄일 수 있다. 전형적인 4X4 delay table 을 적용하는 라이브러리 에서 1 instance 만으로 비교한다면 기존 방식으로 16 회 simulation 이던 것을 새로운 방식 에서는 단 1 회의 A mode simulation 만으로 characterization 이 완료 되므로 **90 % 이상의 소요시간 감소 효과가 있다.** 그러나 무엇보다 characterization 의 생명은 정확성(accuracy) 에 있는 것이다. 따라서 본 논문에서는 0.35 um technology 의 전형적인 two-port SRAM 의 기존 방식에 의한 sample reference data 를 기준으로 새롭게 제안된 방식의 characterization 결과와 비교 검증한 결과 메모리의 중요 timing parameter 에서 3 % 이내의 만족스러운 결과를 얻을 수 있었다. 이 오차는 90 % 이상 줄여든 characterization 시간을 이용하여 inter-extrapolation 이 아닌 실제 simulation 된 메모리 특성 추출 데이터를 90 % 늘릴 수 있으므로 전체 메모리 콤파일러 라이브러리의 characterization 데이터베이스의 획기적인 정확성 증가를 고려한다면 매우 만족스러운 결과라 할 수 있다. 향후 개선사항으로 circuit partitioning 을 위한 아키텍처 제한조건 극복등의 문제를 해결한다면 모든

메모리 라이브러리에 적용할 수 있을 것이다.

참고문헌

- [1] 서 성민, 1998, “아식 셀 라이브러리를 DRAM 설계”, 충남대학교 대학원 석사학위논문.
- [2] Kahng, A. B.; Muddu, S., 1996, “An Analytical Delay Model for RLC Interconnect”, IEEE International Symposium on Circuits and Systems. Vol. issue: No. 4, Page: IV
- [3] Dutta, S; Shetti, S S M; Lusk, S L, Aug.1995, “A Comprehensive Delay Model for CMOS Inverters”, IEEE J. Solid-state Circuits, vol.SC-30,no.8,pp.863
- [4]Alessandro De Gloria, Paolo Faraboschi, and Mauro Olivieri, Jun.1994, ”Design and Characterization of a Standard Cell Set for Delay Insensitive VLSI Design” IEEE Transactions on Circuits and Systems, vol.41, No.6,pp.410-415
- [4] Sharaf, K. M.; Elmasry, M. I., Jan.1994, “An Accurate Analytical Propagation Delay Model for High-Speed CML Bipolar Circuits.”, IEEE J. Solid-state Circuits, vol.SC-29,no.1,pp.31-45
- [6] R.C. Jaeger and R. M.Fox, June 1985, “ Phase plane analysis of the upset characteristics of CMOS RAM cells “, Proc.Univ./Govt./Industry Microelectron.Symp., pp.183-187
- [7] B. Chappell, S.E.Schuster, and G.S.Sai-Halasz, Feb.1985, “Stability and SER analysis of static RAM cells”,IEEE J. Solid-state Circuits, vol.SC-20,no.1,pp.383-390
- [8] K.Anami, M.Yoshimoto, H.Shinohara, Y.hirata, and T.Nakano, Aug.1983, “Design considerations of a static memory cell”, IEEE J. Solid-state Circuits, vol.SC-18,no.4,pp.414-418



참고자료 1 . 사용된 two-port SRAM 1024x4 의 B mode simulation 용 sample circuit block