VLSI 중간 과제 보고서

조 : 8조

2015104027 박정진

2015103999 김윤형

2015104108 장종범

2015104008 김태윤

CONTENTS

[1. Sizing 3](#_Toc41520919)

[1.1 Ideal Sizing 3](#_Toc41520920)

[1.2 Non-Ideal Sizing 4](#_Toc41520921)

[2. Delay 6](#_Toc41520922)

[2.0 Terminology 6](#_Toc41520923)

[2.1 1st order RC Model with decaying exponential (Transient response) 7](#_Toc41520924)

[2.2 Elmore delay of RC Ladder 8](#_Toc41520925)

[2.3 Logical Effort Model 10](#_Toc41520926)

[2.4 Discussion 12](#_Toc41520927)

[3. Corner Analysis 12](#_Toc41520928)

[3.1 Corner Specification 12](#_Toc41520929)

[3.2 Monte Carlo Simulation 14](#_Toc41520930)

[3.3 Inverter VTC 16](#_Toc41520931)

[4. Simulation 19](#_Toc41520932)

[4.1 Inverter 19](#_Toc41520933)

[4.2 NAND2 22](#_Toc41520934)

[4.3 NOR2 26](#_Toc41520935)

[4.4 AND2 30](#_Toc41520936)

[5. Reference 34](#_Toc41520937)

# 1. Sizing

## 1.1 Ideal Sizing

주어진 설계 자료를 참고하면 INVERTER의 경우 (=*β*)는 2, NOR의 경우 4, NAND의 경우는 1로 설계하도록 권장 되어있다. 다음과 같은 *β*로 설계를 진행하는 근본적인 이유는 NMOS와 PMOS의 mobility(; )차이 때문이다. 이동도(mobility)란 전기장(electric field) 속 입자가 갖는 이동속도로 <Figure 1.1>에서 알 수 있듯이, T=300K 일 때 실리콘(silicon)에서 전자(electron)의 이동도(mobility)는 이고 정공(hole)의 이동도(mobility)는 이다.

A screenshot of a social media post

Description automatically generated

<Figure 1.1>[1]

즉, 만일 MOSFET 구조에서 를 만족하고 에 의해 전기장이 생긴다면, 채널(channel)속 전하(electric charge)가 이동을 할 것이며 이 때 전하의 종류에 따라 이동속도가 달라지게 될 것이다. 낮은 전하의 이동속도는 낮은 전류( )를 의미한다.

따라서 회로가 동일한 rise, fall delay()를 갖게 만들기 위해선 전류 식의 변수를 조절하여야 한다. 전류와 변수간 관계는 아래와 같다.

이 때, 는 공정변수(Fabrication factor)로 같은 material로 이뤄진 die 위에서 를 변경하는 것은 매우 복잡할 것이다. 즉, 수정 가능한 변수는 가 되지만 은 보통 공정의 최소 길이(minimum length)로 고정함으로 실제 를 일치시키기 위해 의 비를 조절하는 것이 보편적이다. 실제 본 프로젝트를 진행함에 있어 **Length(를 50nm로 고정시킨 이유는 layout에 사용하는 freePDK model이 지원하는 minimum length가 50nm이기 때문이다.**

계산의 편리성을 위해 **라 가정한다면 단일 NMOS와 PMOS가 동일한 전류 즉, 동일한rise, fall delay 갖기 위해선 hole을 전하로 갖는 PMOS의 는 electron을 전하로 갖는 NMOS의 의 2배가 되야 할 것이다.** 이와 같은 이유로 Inverter 회로에서 pull-up쪽 PMOS와 pull-down쪽 NMOS의 ‘width ratio’가 2:1 로 설계가 된 것이다.

다만 의 갖게 설계된 inverter 회로에 대해 hspice simulation돌려보면 가 일치하지 않음을 확인 가능했다. 이는 크게 두가지 이유로 설명이 가능한데 첫 번째, electron mobility(와 hole mobility((가 정확히 2:1의 비를 갖지 않는다는 점이 있고, 두 번째로는 우리가 알고 있던 MOSFET의 전류식이 실제 MOSFET 동작을 완벽히 나타내지 않는 간략화 된 식이라는 점이다.

A picture containing bird

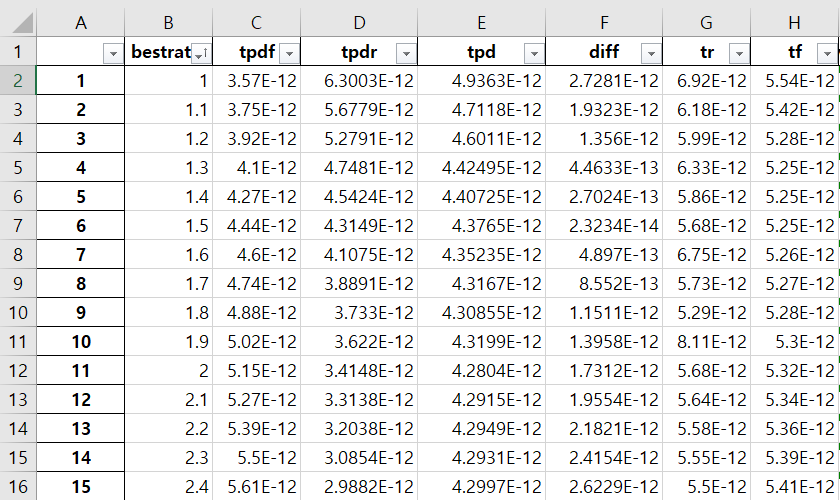
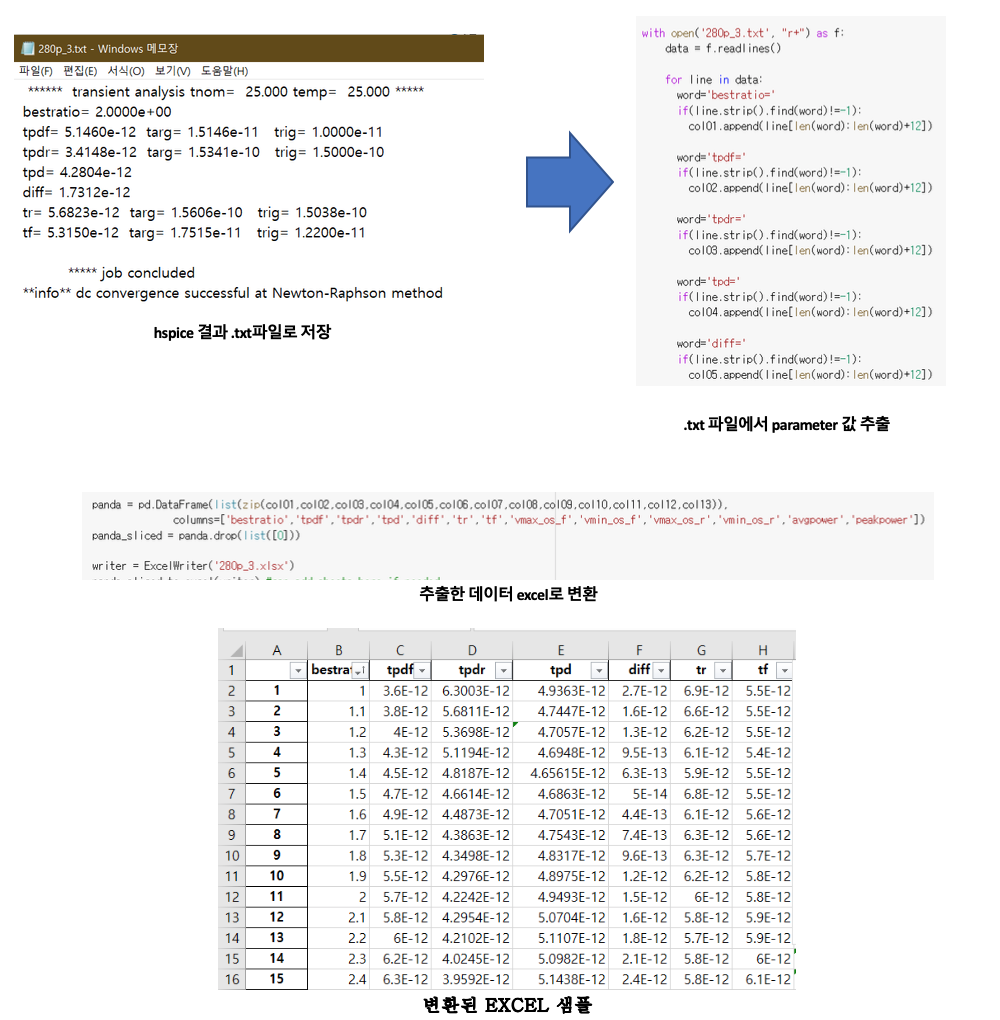
Description automatically generated

Non reliable I-V equation for Cadence

즉, 실제 MOSFET의 동작을 수치해석한 여러가지 모델들(Level 1 ~74)은 다양한 Parameter에 근거해 전류 값을 계산할 것이고 그에 따라 값은 단순히 mobility 변수 하나만으로 전류가 linear하게 바뀌지 않을 것이다.

## 1.2 Non-Ideal Sizing

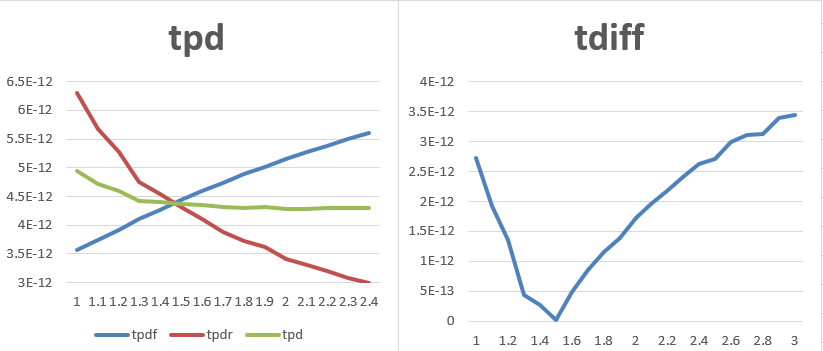
본 프로젝트에서 사용하는 NMOS PMOS model은 MOFET의 여러가지 모델 중 BSIM4(Berkeley Short-channel IGFET model)(LEVEL=54) model로 해당 조건에서 inverter 회로가 가 되기 위한 와 가 최소가 되기 위한 를 구해 보았다. 를 결정하기 위해 hspice의 optimization 기능과 Sweep 기능을 이용하였다. SWEEP을 진행할 때 주의해야 할 점으론 입력신호(input signal)의 rising time과 falling time을 너무 길게 잡으면(ex. 20n) MOSFET의 동작 능력을 제대로 비교하기 어렵다는 점이다. 따라서 시뮬레이션은 20ps의 rise delay와 fall delay를 갖는 입력신호로 하였고 Width를 1.00부터 4.00까지 0.01의 delta()로 PARAM SWEEP을 진행하였다



<Figure 1.2>

이렇게 나온 HSPICE 결과를 python으로 처리하여 시각화 하였는데 그 방법은 <Figure 1.2>와 같다. Python의 Pandas library 기능 중 ‘ExcelWriter’을 이용하였으며 hspice 결과로 나온 txt 확장자 파일을 입력으로 받고 각 줄에서 내가 지정한 parameter name을 찾는 방식으로 구현하였다.

결과로 나온 excel을 차트화 시킨 결과는 아래와 같았다.



<Figure 1.3>

와 차트를 보면 중간중간 불완전한 OPTION 설정에 따른 Noise가 관찰되지만 차트 속 곡선 경향을 살펴보면 대략 이론상의 곡선 경향(<Figure 1.4>)과 비슷하게 나온 것을 확인 가능했다.

A close up of a map

Description automatically generated

<Figure 1.4>[2]

실제 가 2, min\_diff, min\_average\_delay를 만족할 때를 비교해보면 <Table 1.1>와 같다.

<Table 1.1>

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | 1.87 | 1.52 |
|  | 5.146 | 4.9783 | 4.4218 |
|  | 3.4148 | 3.4445 | 4.4349 |
|  | 4.2804 | **4.2114** | 4.42835 |
|  | 1.7312 | 1.8358 | **0.013032** |

결과적으로 inverter에서 를 2로 설정한 것보다 더 1.52로 설정했을 때 가 더 작음을 관찰 가능했다. 또한 가 1.87일 때에 평균 delay인 가 가장 작음을 알 수 있었다. 는 일 때보다 PMOS의 크기가 작을 것이며, 이는 MOSFET이 낮은 power consumption를 갖는 것을 의미한다. 하지만 때가 일 때 보다 가 크다는 것도 알 수 있었다. delay간의 차이가 클수록 hot electron degradation과 같은 문제가 발생할 것이며, 회로의 동작점이 낮아져 회로의 noise margin을 감소시킬 것이다.

<Figure 1.5>는 hspice의 DC simulation 기능을 이용해 에 따른 Voltage Transfer Curve(VTC)를 나타낸 것이다. 실제로 가 1.52인 경우(빨간색)의 동작점이 894.41mV로 가 1.87인 경우(노란색)와 가 2인 경우(파란색)보다 낮음을 확인할 수 있었다. 또한 낮은 동작점으로 작동할수록 낮은 를 가져 Noise margin이 작을 것을 시각적으로 관찰 가능했다.

A close up of a computer

Description automatically generated

<Figure 1.5>

따라서, 회로를 설계함에 있어 단순히 나가 sizing의 결정요소가 되서는 안되고 area, power, reliability 같은 요소들을 따져서 를 결정해야한다. 본 프로젝트에서는 이와 같은 정보들을 알고 있지만 편의상 Inverter의 를 2로 두고 설계를 진행하였다.

# 2. Delay

1. Sizing에서 언급했듯이 P:N=2:1인 Inverter로 설계를 하였고 그것을 기준으로 Delay에 대해 설명하겠다. Delay Analysis에는 여러가지 방법이 있는데 고전적 방법인 RC model delay, Elmore Delay of RC Ladder그리고 logical effort를 이용한 방법이 있다. 이 3가지 방법 모두 linear delay model이다. 각각의 방식들이 어떻게 Delay를 계산하는지 그리고 hand-calculation을 진행할 때 왜 logical effort 모델을 사용하는지를 통해 기본적인 gate들 (NAND, NOR...)의 size선정 이유를 알 수 있다.

## 2.0 Terminology

: maximum rising propagation delay

-from input to rising output crossing

: maximum falling propagation delay

-from input to falling output crossing

: average propagation delay

- +

: rise time

- from output crossing

: fall time

- from output crossing

: minimum rising contamination delay

-from input to rising output crossing

: minimum falling contamination delay

-from input to falling output crossing

: average contamination delay

- +

## 2.1 1st order RC Model with decaying exponential (Transient response)

A close up of text on a white background

Description automatically generatedA close up of text on a white background

Description automatically generated

<Figure 2.1>[2] <Figure 2.2>[3]

<Figure 2.1>과 <Figure 2.2>를 통해 다음을 알 수 있다.

( : internal capacitance,  : load capacitance)

이를 통해 Delay Formula를 세우면

with (

(effective fanout)

RC delay model은 Inverter같이 간단한 회로에서는 괜찮지만 1차를 넘어 2, 3차, 고차로 갔을 때는 식이 매우 복잡 해진다. 만일 RC가 Ladder 형태로 추가되면 추가된 만큼 n차 response로 계산을 해야 하지만 hand-calculation의 한계로 n차를 1st-order response로 approximate해야 한다. 이는 부정확한delay 결과의 얻게 됨을 의미한다. 또한 1st-order single time constant는 intermediate nodes의 behavior를 설명하기에는 부족하다. RC delay 모델의 계산의 복잡함과 부정확함을 극복하기 위해 Elmore delay model을 사용한다

## 2.2 Elmore delay of RC Ladder

RC delay 모델이 NMOS, PMOS의 absolute value를 이용했다면, Elmore Delay 모델은 R, C의 정의를 다르게 해 relative value로 Delay를 계산한다. 그 결과 RC delay의 방식보단 더 쉬운 delay estimation 과 낮은 error rate를 보인다. 또한 해당 모델은 이후 logical effort를 이해하는데 도움이 된다.

Definition

: effective resistance (, depends on average current as gate switches)

(A unit NMOS transistor is defined to have effective resistance R)

: gate capacitance of a unit transistor of either favor(, neglect )

: electrical effort = fanout =

Hand-Calculation

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

<Figure 2.3>[4] <Figure 2.4>[4]

<Figure 2.3> 과 <Figure 2.4>을 통해 Inverter, NAND2, NOR2에 대해 분석해보면 다음과 같다.

A close up of text on a white background

Description automatically generated

A close up of text on a white background

Description automatically generated

A close up of text on a white background

Description automatically generated

<Table 2.1>

|  |  |  |  |
| --- | --- | --- | --- |
|  | INV | NAND2 | NOR2 |
|  | (3+3h)RC | (6+4h)RC | (8+5h)RC |
|  | (3+3h)RC | (7+4h)RC | (6+5h)RC |
|  | (3+3h)RC | (6.5+4h)RC | (7+5h)RC |
|  | - | (3+2h)RC | - |
|  | - | - | (3+2.5h)RC |

Elmore delay model로 분석한 결과를 정리하면 <Table 2.1>과 같다. 하지만 Elmore delay 모델도 size를 알아야 한다는 단점이 있다. 또한 gate가 여럿 있을 때 분석하기 힘들다는 점과 비교가 logical effort에 비해 빠르게 되지 않는다는 단점도 존재한다. 이러한 단점을 해결하기 위해 Logical Effort Model을 사용한다.

## 2.3 Logical Effort Model

Logical Effort 모델은 다른 모델에 비해 매우 간단하고, back-of-envelope[[1]](#footnote-1)를 가능하게 한다. 또한 회로간 비교가 수월하며 hand-calculation이 다른 모델에 비해 상대적으로 빠르다

Definition

( )

: normalized delay

: absolute delay ()(typical)

: inverter time constant

즉, inverter를 기준으로 하여 모든 delay를 normalized시켜 비교가 쉽게 되도록 하는 것이다.

Formula를 사용해 <Table 2.1>을 정리하면 <Table 2.2>이 만들어진다.

Hand-Calculation

|  |  |  |  |
| --- | --- | --- | --- |
|  | INV | NAND2 | NOR2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

<Table 2.2>

위의 delay 식을 component 별로 나눠서 써보면

*:* effort delay (=(stage effort)

: logical effort

: electrical effort(fanout) =

: parasistic delay

먼저 parasitic delay를 분석해보면 정의가 부하를 제외한 gate 동작에서 발생하는 delay이다. 즉 h=0 일 때 (output에 연결된 load cap, gate가 없을 때) delay 값이다. Inverter가 1로 기준이다. 이 말을 달리 해석하면 Inverter는 모든 logic gate중 최소의 delay를 가지고 있다는 뜻이 된다.

위 <표 3-1>을 보면 contamination delay를 볼 때 h=0 인 경우(intrinsic delay) NAND2, NOR2또한 delay가 1이 됨을 알 수 있다. delay가 같다는 말은 drive되는 output current양이 같다고 볼 수 있다. 결론적으로 logic gate의 logical effort[[2]](#footnote-2)는 단어 뜻 그대로 logic gate 마다 각자 갖고 있는 logic의 결과를 inverter와 똑같은 성능(똑같은 load에서 똑같이 구동)으로 내기 위해 얼만큼 노력을 더 해야 하는가를 의미한다.

이를 통해 우리는 왜 NAND2와 NOR2의 Sizing에 대한 이해를 다시 한번 할 수 있다. NAND2 : P/N = 2/2, NOR2 : P/N = 4/1 을 잡아야 contamination delay(best case)가 logic gate의 최소 intrinsic delay(Inverter intrinsic delay=1)와 같아질 수 있는 것이다.

Maximum propagation delay path(worst-case)를 생각해봐도 마찬가지이다. Inverter와 어떤 logic gate의 Pull up network, Pull down network에서 각각 보이는 저항을 같게 설계함으로 best case에서 생각했던 논리의 역으로, Inverter와 drive되는 ouptut current가 같으니, 구동성능 즉, delay가 같게 된다는 의미이다. 그래서 이 경우에도 마찬가지로 NAND2 : P/N = 2/2, NOR2 : P/N = 4/1 로 잡아야 Pull up network, Pull down network에서 보이는 저항이 같아지게 된다.

<Table 2.2>의 delay formula를 살펴보면 의 형태로 표현되어 있으므로 우리는 손쉽게 각 logic gate의 g와 p를 알 수 있다.

<Table 2.3>

|  |  |  |  |
| --- | --- | --- | --- |
|  | INV | NAND2 | NOR2 |
|  |  |  |  |
|  |  |  |  |

이와 같이 gate들의 특성을 numerical characterize하여 나타나므로 각 게이트의 delay를 손쉽게 비교, 또한 fast circuit 이 무엇인지에 대해 이야기할 수 있다. 이 logical effort의 단점은 대표적으로 interconnect, 즉 wire를 이용한 디자인에서 iteration 계산이 필요하다는 점과 constrained delay에 대한 최소의 area, power에 대한 정보를 제공하지 않는다는 점이다. 따라서 Logical effort는 회로의 maximum speed를 측정할 때만 유리하다.

## 2.4 Discussion

지금까지 설명한 linear delay model에는 simple한 delay estimation을 가능하게 하지만, 당연하게도 몇 가지 limitation이 존재한다. 첫 번째로는 input source에 관련된 문제이다. linear delay model에서 사용하는 input source는 ideal한 pulse 혹은 step function을 가정하며, input에 rise, fall time이 존재하지 않는다. 하지만 실제에서는 rise, fall 이 0인 source는 없으며 무조건 slope를 가지고 있다. 그 결과, input의 rise, fall time에 의해 propagation delay가 예상했던 것 보다 더 길게 측정이 된다. input source의 또 다른 문제는mutiple-input gate에서 하나가 switching 될 때 다른 것들은 모두 stable하게 있다고 가정을 한다는 것이다. 하지만 실제로는 series transistor의 input이 동시에 ON이 되면 delay는 예상보다 더 길어지게 된다. 그 이유는 series transistor가 transition 초기에는 부분적으로(순차적으로) ON이 되기 때문이다. 반대의 경우 parallel transistor 경우 동시에 ON이 되면 delay가 예상보다 더 짧아지게 되는데, parallel transistor 모두 current가 흐르게 되기 때문이다. 게다가 delay는transistor의 input pattern에 따라서도 달라진다.

두 번째로는 velocity saturation의 문제이다. long-channel에서는 R ~ L로 current가 L에 linear하게 감소하였다. 하지만 transistor가 fully velocity-saturated되면 I와 R이 L에 independent하게 된다. 이는 series transistor에서이 계산 상의 보다 작게 된다. 이 현상은 NMOS가 PMOS보다 더 두드러지게 나타나는데 그 이유는 NMOS의 mobility가 더 크고 velocity saturation 정도가 더 강하기 때문이다. 결론적으로 실제에선 series transistor의 이 더 작아지게 되므로 delay는 예상보다 작게 나온다. 따라서 logical effort에서 inverter와 same current를 drive하기 위해선 size를 다르게 설계해야 한다.

(, saturated voltage)

위의 식을 통해 series sizing 비율을 정해야 하며 이 비율은 공정마다 다른 값을 가질 것이다. 전반적으로delay를 예상한 값에 맞추려면(delay가 짧은 것(series)을 길게(inverter)) series 연결된 부분을 기존 logical effort를 계산했을 때 나온 size보다 더 작게 설계해야 할 것이다.

세번째 이유는 capacitance의 차이이다. Bootstrapping으로 불리는 effect는 gate들이 연결 되 있을 때 나타나는데, 두 gate의 input 과 output 사이에 존재하는 coupling capacitor때문이다. Miller effect로도 알려져 있는 이 현상은 디지털에서는 두드러지는 effect는 아니지만 영향이 있다는 것은 부정할 수 없다. 우리는 C = 로 계산을 하였지만, 실제 동작 영역에선 값이 합쳐져 두 배가 된다. (C = ) 그 이유는 transistor가 partially ON, OFF 즉 linear region에 있을 때 가 회로의 gain이 곱해진 값으로 증가되기 때문이다. 그래서 원래 무시했던 값이 무시할 수 없을 정도로 커지며 delay가 예상했던 것 보다 더 커지게 되는 현상을 볼 수 있다.

# 3. Corner Analysis

Manufacturing단계에서는 많은 variation이 일어나는데 모두 회로 동작에 영향을 미치는 요소들이다. Process 파트에선 threshold, channel length, interconnect dimensions factor가 있고 Environment에선 Voltage, Temperature variation이 존재한다. 뿐만 아니라 Transistor가 switching 될 때마다 Aging/Wear-out을 일으키며, 이는Hot carrier effect, NBTI(Negative Bias Temperature Instability), TDDB(Time-Dependent Dielectric Breakdown) 등 hard error들을 일으키며 안정도 부분에서 안 좋은 결과를 내게 된다. 이러한 이유로 initial design이 완성되면 이런 variation parameter의 extreme value에서 simulation을 통해 worst-case 검증을 해야 한다.

Circuit Designer들은 Process, Environmental variation을 고려하는데, 5개의 circuit component(NMOS, PMOS, Wire, VDD, Temp)를 3개의 letter(T-Typical, S-Slow, F-Fast)로 나누어 simulation을 진행한다. 이에 따라 본 보고서에서는 먼저 Corner들이 갖는 특성들을 Specification Table, VTC 등을 이용하여 각 코너에 따른 특징들을 파악하려 한다. Corner는 Wire의 모델을 정할 수 없어 4개 Temp, VDD, NMOS, PMOS만을 지정했으며, 코너의 letter 표시는 Temperature, VDD, NMOS, PMOS 순이다[[3]](#footnote-3).

## 3.1 Corner Specification

각 코너에 따라 알아 봐야할 중요한 포인트들인 NMOS&PMOS ON current, Gate Leakage Current, OFF Current, Threshold Voltage, Effective Length를 시뮬레이션 하였다.

MOSFET 모델은 freePDK\_45nm를 사용하였으며, 그중 low threshold voltage model인 VTL을 사용했다. freePDK\_45nm는BSIM4(LEVEL=54) 모델이다.

우리가 대표적으로 사용할 TTSS, TTTT, TTFF의 Corner Specification은 <Table 3.1>과 같다.

<Table 3.1>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | NMOS | | | PMOS | | |
|  | Slow | Typical | Fast | Slow | Typical | Fast |
|  | 1266 | 1350.8 | 1428.3 | 859.66 | 912.07 | 967.03 |
|  | 11881 | 21836 | 42843 | 9376.8 | 18590 | 36380 |
|  | 1374.5 | 2280.4 | 4337.9 | 2122.3 | 3831.7 | 6888.4 |
|  | 0.14762 | 0.11265 | 0.079866 | 0.14419 | 0.11017 | 0.075155 |
|  | 0.2338 | 0.20644 | 0.17878 | 0.25617 | 0.22847 | 0.20054 |
|  | 23.55 | 22.50 | 21.45 | 23.55 | 22.50 | 21.45 |

은 constant-current method를 사용하여 구한 것이다. MOSFET이 점점 짧아짐에 따라 Short-channel effect를 비롯한 non-ideal character들이 많아지고 기존의 Extrapolation Method는 실제 Die에서 측정한 와 차이를 보였다. 이는 velocity saturation의 영향이 증가함에 따라 Square law가 더 이상 적용되지 않기 때문이다. 또한 MOSFET의 구조가 시간이 흐르며 급속도로 바뀌는 반면, Extrapolative model은 그 바뀐 모델의 를 계산하는 식을 다시 짜려면 시간이 많이 필요했다. 그래서 circuit designer가 정한 에 대응하는 를 구하는 방식인 CC Method(Constant-Current Method)가 주류가 되며 를 구하는 방법은 MOSFET Technology와 같이 성장할 수 있게 되었다.

이 방법의 단점은 <Figure 3.3>에서 알 수 있듯이 이 user/process-specific이어서 가 에 strongly dependent하다는 점이다. 본 프로젝트에서는 가장 보편적으로 쓰이는 값인 로 지정하고 측정하였다.

A screenshot of text

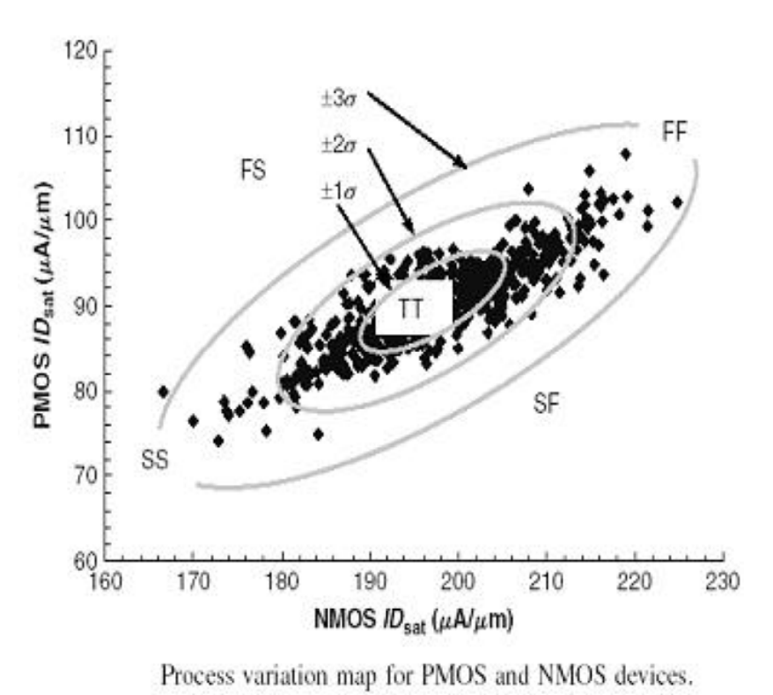
Description automatically generated

<Figure 3.3>[8]

<Table 3.1>을 통해 알 수 있듯이 Fast>Typical> Slow 순으로 가 커졌다. 이 큰 것을 통해 Delay Time이 적을 것을 예측할 수 있으며, 또한 를 통해 leakage power가 클 것임을 알 수 있고 그에 따른 power consumption이 많을 것을 예측할 수 있다.(이 큰 것 또한 영향을 끼친다) 또한 Threshold voltage의 크기는 Slow>Typical>Fast 순서를 갖었다. 이를 통해 Transition에서 Fast가 가장 빠르게 전류가 흐를 것으로 파악되며, 이는 Transition에 대한 반응이 빠를 것임을 암시한다.

## 3.2 Monte Carlo Simulation

실제로 Fabrication이 corner value에 맞게끔 static하게 제작되지 않으므로 Monte Carlo Simulation을 통해 Design Checking을 진행한다.



<Figure 3.1>[6]

Monte Carlo를 진행하는 또다른 이유는 Fixed Corner Analysis가 pessimistic 하여, unrealistic, unphysical한 extreme combination으로 overdesign 되게 만들 수 있다는 점이다.

Monte Carlo Simulation을 하기 위해서는 Statistical Model이 필요한데, 대부분의 Foundry에서는 이 Model을 제공한다. 하지만 없는 경우에도 모델을 세워서 할 수 있는데 다음 식에 들어갈 parameter value를 알아야 한다.

: Hspice absolute gaussian[6]

: Typical Library's parameter value

: Fast Library's parameter value

: Slow Library's parameter value

: Sigma Difference (typically, 3)

Slow, Fast, Typical의 모델 parameter들 중 각 코너에 따라 바뀌는 값을 정리해보면 <Table 3.2>과 같다.

<Table 3.2>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | nMOS | | | pMOS | | |
|  | slow | typical | fast | slow | typical | fast |
| TOXREF | 1.17E-09 | 1.14E-09 | 1.1E-09 |  |  |  |
| TOXE | 1.17E-09 | 1.14E-09 | 1.1E-09 | 1.3E-09 | 1.26E-09 | 1.22E-09 |
| TOXM | 1.17E-09 | 1.14E-09 | 1.1E-09 | 1.3E-09 | 1.26E-09 | 1.22E-09 |
| XJ | 2.05E-08 | 1.98E-08 | 1.9E-08 |  |  |  |
| NDEP | 3.6E+18 | 3.4E+18 | 3.4E+18 |  |  |  |
| VTH0 | 0.347 | 0.322 | 0.297 | -0.327 | -0.302 | -0.277 |
| CF | 1.283E-10 | 1.289E-10 | 1.297E-10 | 1.26E-10 | 1.267E-10 | 1.274E-10 |
| LINT | 3.225E-10 | 3.75E-10 | 4.275E-10 | 3.225E-10 | 3.75E-10 | 4.275E-10 |

이 value 들을 이용하여 HSPICE Monte Simulation을 이용하면 되는데 안타깝게도, NP Correlation 값을 알지 못하면 <Figure 3.2>처럼 unphysical combination 결과가 도출된다[7]. <Figure 3.3>은 FreePDK를 Variation, Correlation 없이 Monte Simulation을 돌렸을 때 나온 plot이다.

A close up of a map

Description automatically generated

<Figure 3.2>[7]

A screenshot of a cell phone

Description automatically generated

<Figure 3.3>

보통 Foundry 회사에서 NP Correlation 값을 주지만, 우리는 교육용 라이브러리를 사용하기에 이 값을 알 수 없다. 하지만 만약 아는 경우 다음 식을 통해 Monte Carlo Simulation을 진행할 수 있다.

본 보고서에서는 NP Correlation = 0.5으로 임의로 잡고, Monte Simulation을 진행하였다. <Figure 3.4>는 그 결과 plot이다.

A close up of a map

Description automatically generated

<Figure 3.4>

## 3.3 Inverter VTC

Temperature

빨강(FTTT) : 노랑(TTTT) : 초록(STTT) :

A close up of a device

Description automatically generated 

<Figure 3.5> <Figure 3.6>

<Figure 3.5>, <Figure 3.6>을 보면 알 수 있듯이 Fast > Typical > Slow 순으로 VTC Curve의 slope가 가파른 것을 확인할 수 있다. 즉, 더 빠르게 switching 된다는 의미이다.

TTSS vs TTTT vs TTFF

빨강(TTTT), 노랑(TTSS), 초록(TTFF)A close up of a device

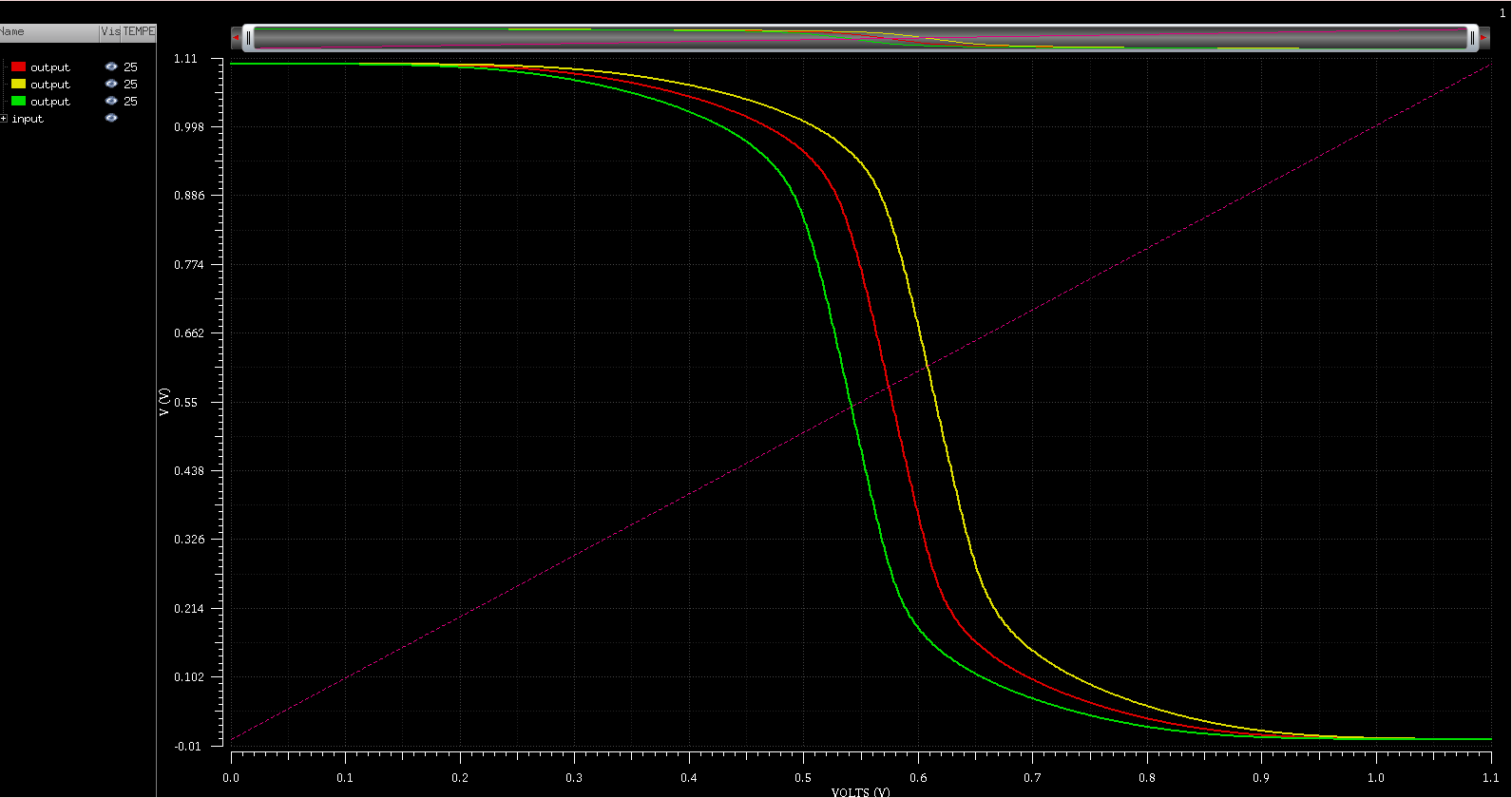
Description automatically generated

<Figure 3.7>

Temperature 비교와 마찬가지로 TTFF > TTTT > TTSS순으로 VTC Curve의 slope가 가파른 것을 확인할 수 있다.

TTSF vs TTTT vs TTFS

빨강(TTTT), 노랑(TTSF), 초록(TTFS)



<Figure 3.8>

A picture containing monitor, computer

Description automatically generatedA close up of text on a white background

Description automatically generated<Figure 3.9> <Figure 3.10>[2]

시뮬레이션 결과 <Figure 3.10> 과 같은 형태의 그래프가 나왔다. Output voltage = Input voltage 의 그래프인 분홍 선 기준으로 위는 NMOS(falling), 아래는 PMOS(rising) 가 동작하는 것인데 시뮬레이션 결과와 같이 일 때 Fast NMOS인 초록이 가장 빨리 에 도달하고 일 때는 Fast PMOS인 노랑이 가장 빨리 에 도달한다.

# 4. Simulation

본 시뮬레이션에서는 input signal의 transition time을 너무 길게 잡으면 동작 능력을 제대로 비교하기 어려우므로, transition time을 로 두고 하였다.

## 4.1 Inverter

Schematic & Layout

점수판이(가) 표시된 사진

자동 생성된 설명

A picture containing light, red, hanging, traffic

Description automatically generatedA close up of a sign

Description automatically generated

<Figure 4.1>

DRC & LVS

A screenshot of a cell phone

Description automatically generatedA screenshot of a video game

Description automatically generated

Pre-Simulation 전자기기, 컴퓨터, 앉아있는, 대형이(가) 표시된 사진

자동 생성된 설명

<Figure 4.2>

Pre-Sim Corner

빨강(TTTT), 노랑(TTFF), 초록(TTSS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명 컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.3> <Figure 4.4>

Pre-Sim vs PLS (TTTT)

빨강(Pre-Sim), 초록(PLS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명 컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.5> <Figure 4.6>

Device Characteristics Table

<Table 4.1>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TTSS | | TTTT | | TTFF | |
| Pre-Sim | PLS | Pre-Sim | PLS | Pre-Sim | PLS |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| ( |  |  |  |  |  |  |

Discussion

Delay Time을 정리해서 보았을 때 이전 NMOS, PMOS 코너의  값에서 봤듯이 값이 크면 Delay가 작은 결과를 보여준다. 그래서 FF > TT > SS 의 순서로 Delay Time이 짧다. 하지만 등 흐르는 전류 값이 크므로 Power consumption 또한 FF > TT > SS순으로 큰 것을 확인 할 수 있다.

아래에서 NAND2, NOR2, AND2의 Delay값이 합리적인지 비교하기 위해 Inverter값을 기준으로 잡을 것이다. 프로젝트 설계사항에 Load가 없으므로 fanout = 0 일 것이다. 즉 normalized delay  로 한다. (Pre-Sim, TTTT) Pre-Sim, PLS 결과도 Layout에 존재하는 Parasitic Resistance, Capacitance의 존재를 확인할 수 있다. <Figure 4.6>과 <Table 4.1>을 통해 알 수 있듯이 PLS가 Parasitic한 요소들로 인해 RC Time constant 가 커져 Delay가 Pre-Simulation보다 더 큰 것을 관찰할 수 있다.

## 4.2 NAND2

Schematic & Layout

작은, 검은색, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

A close up of a red light

Description automatically generatedA picture containing red, sitting, pole, meter

Description automatically generated

<Figure 4.7>

DRC & LVS

A screenshot of a cell phone

Description automatically generatedA screenshot of a video game

Description automatically generated

Pre-Simulation 전자기기, 컴퓨터, 회로이(가) 표시된 사진

자동 생성된 설명

<Figure 4.8>

Pre-Sim Corner

빨강(TTTT), 노랑(TTFF), 초록(TTSS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명

컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.9>

Pre-Sim vs PLS (TTTT)

빨강(Pre-Sim), 초록(PLS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명

모니터, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.10>

Device Characteristics Table

<Table 4.2>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TTSS | | TTTT | | TTFF | |
|  | Pre-Sim | PLS | Pre-Sim | PLS | Pre-Sim | PLS |
|  |  | 12.001 |  | 11.438 |  |  |
|  |  | 9.5320 |  |  |  |  |
|  |  | 10.766 |  |  |  |  |
|  |  | 5.0869 |  | 4.5956 |  | 4.1446 |
|  |  | 8.5438 |  | 8.0170 |  | 7.5603 |
|  |  | 9.8333 |  | 9.5527 |  |  |
|  |  | 10.191 |  | 9.8449 |  |  |
|  |  | 116.04 |  | 135.90 |  |  |
| ( |  | 126.94 |  | 120.96 |  |  |

Discussion

이전에 Inverter에서 나온 결과를 사용하기로 했기에 로 잡았다. <Table 2.2> 에서 알 수 있듯이 NAND2의 delay는 이론적으로 가 나와야 할 것이다. (h=0) 하지만 시뮬레이션 결과의 normalized delay를 구하면 로 예상보다 낮은 결과가 나온다. 이러한 차이를 보이는 이유는 Delay파트에서 설명했듯이 크게 두 가지가 대표적이다. 첫 번째로 NAND2의 의 차이다. NMOS는 Series로 연결되어서 이론적으로 이 되어야한다. 즉, 가 되어야 할 것이다. 하지만 시뮬레이션 결과는 으로 더 작은 값인데 그 이유는 velocity saturation으로 인해 R이 L에 independent하므로 그렇다. 그렇기 때문에 이 예상했던 것보다 더 작아지게 되므로 delay는 예상보다 빠르게 나온다.

두 번째로는 의 차이이다. 이론적으로 NAND2의 Best-Case Parasitic Delay(=1)는 Inverter의 Parasitic Delay(=1)와 같아야 한다. 하지만 Inverter의  이지만, NAND2의 이다. 이런 결과가 나오는 이유는 Best-Case인 경우 Parallel PMOS가 동시에 ON이 되어 전류가 더 흐르게 된다. 따라서 Delay가 예상보다 짧아지게 되는 것이다. 만약 두 개의 Parallel PMOS가 동시에 ON이 되었을 때 전류 값이 대략 2배 정도 커진다 생각하면 시뮬레이션 결과 은 해당 이유에 대한 합리적인 결과임을 알 수 있다.

## 4.3 NOR2

Schematic & Layout

전자기기, 회로, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

A picture containing light, clock, sitting, black

Description automatically generated

<Figure 4.11>

DRC & LVS

A screenshot of a cell phone

Description automatically generatedA screenshot of a video game

Description automatically generated

Pre-Simulation

전자기기, 컴퓨터, 회로이(가) 표시된 사진

자동 생성된 설명

<Figure 4.12>

Pre-Sim Corner

빨강(TTTT), 노랑(TTFF), 초록(TTSS)

전자기기, 컴퓨터이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명



<Figure 4.13>

Pre-Sim, PLS (TTTT)

빨강(Pre-Sim), 초록(PLS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명

컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.14>

Device Characteristics Table

<Table 4.5>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TTSS | | TTTT | | TTFF | |
| Pre-Sim | PLS | Pre-Sim | PLS | Pre-Sim | PLS |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | 8.4167 |  | 7.7943 |  | 7.3123 |
|  |  | 9.3370 |  | 8.7312 |  | 8.2199 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| ( |  |  |  |  |  |  |

Discussion

NAND2에서 분석한 방법과 똑같이 적용했을 때NOR2의 delay는 이론적으로 임을 알 수 있다.(h=0) 하지만 시뮬레이션 결과의 normalized delay를 구하면 의 결과가 나온다. NOR2는 NAND2와 조금은 경향이 다른 결과 값이 나온다. 먼저 이렇듯 을 보면 NAND2에서 말했던 것처럼 PMOS는 Series로 연결되어서 이론적으로 이 되고 Delay는 두 배가 되어야 한다. () NAND2의 결과와 다르게 로 이론과 유사한 결과가 나온다. NAND2와 NOR2의 결과의 경향이 다른 이유는 Series로 연결 되어있는 MOSFET이 N-Type인지 P-Type인지 다르기 때문이다. 위에서 언급한 Velocity Saturation에 의한 R이 L에 independent해지게 되는 현상은 PMOS가 NMOS보다 덜 하게 나타나는데 그 이유는 NMOS의 mobility가 더 크고 velocity saturation 정도가 더 강하기 때문이다. 그렇기 때문에 NAND2는 이론 값과 차이가 크지만 NOR2는 그 차이가 덜 한 것이다.

## 4.4 AND2

Schematic & Layout

텔레비전, 화면, 모니터, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

A sign in the dark

Description automatically generatedA picture containing light, black, red, hanging

Description automatically generated

<Figure 4.15>

DRC & LVS

A screenshot of a cell phone

Description automatically generatedA screenshot of a video game

Description automatically generated

Pre-Simulation

검은색, 앉아있는, 컴퓨터, 대형이(가) 표시된 사진

자동 생성된 설명

<Figure 4.16>

Pre-Sim Corner

빨강(TTTT), 노랑(TTFF), 초록(TTSS)

컴퓨터이(가) 표시된 사진

자동 생성된 설명전자기기, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.17>

Pre-Sim, PLS (TTTT)

빨강(Pre-Sim), 초록(PLS)

컴퓨터, 모니터, 노트북이(가) 표시된 사진

자동 생성된 설명컴퓨터이(가) 표시된 사진

자동 생성된 설명

컴퓨터이(가) 표시된 사진

자동 생성된 설명

<Figure 4.18>

Device Characteristics Table

<Table 4.5>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TTSS | | TTTT | | TTFF | |
| Pre-Sim | PLS | Pre-Sim | PLS | Pre-Sim | PLS |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | 13.660 |  | 12.565 |  | 11.450 |
|  |  | 17.114 |  | 15.829 |  | 14.732 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| ( |  |  |  |  |  |  |

Discussion

AND2는 사실 상 multistage여서(NAND2-INV) Delay 계산을 다음과 같이 해야한다.

,

여기서는 간단히 branch effort( 가 1이고 load cap 이 없으므로 Delay가 Parasitic Delay의 합으로만 구성 되어있다고 생각한다. AND2의 Delay 정의는 다음과 같이 했다.

<Table 4.5>를 보면 알 수 있듯이 AND2 또한 stage 잇는 interconnect의 추가 delay를 고려한다면 예측했던 이론 값과 유사한 값을 나타내는 것을 볼 수 있다.

# 5. Reference

[1] Donald A. Neamen, *Semiconductor Physics and Devices*, McGRAW Hill India, 2013.

[2] Jan M. Rabaey, Anantha Chandrakasan, Borivoje NiKolic, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall India, 2002.

[3] Vishal Sharma, "Rc delay in vlsi", Oct 2018 URL: <https://www.slideshare.net/VishalSharma754/rc-delay-modelling-in-vlsi>

[4] Neil Weste, David Harris, *CMOS VLSI Design: A Circuit and System Perspective*, Pearson, 2011.

[5] Kerwin Khu, "Statistical Modeling for Monte Carlo Simulation using Hspice", *SNUG*, Singapore, 2006.

[6] Ban P. Wong, Anurag Mittal, Yu Ca0, and Greg Starrr, "Nano-CMOS Circuit and Physical Design", *Wiley* 2005.

[7] S. Inohira, T. Shinmi, M.Nagata, T. Toyabe, K. Iida, "A Statistical Model Including Parameter Matching for Analog Integrated Circuits Simulation", *IEEE Trans. Computer-Aided Design*, Vol. 4, Oct 1985.

[8] Alvin Loke, Zhi-Yuan Wu, Reza Moallemi, Dru Cabler, Chad Lackey, Tin Tin Wee, Bruce Doyle "Constant-Current Threshold Voltage Extraction in HSPICE for Nanoscale CMOS Analog Design", *SNUG*, San Jose, 2010.

1. A back-of-the-envelope calculation is a rough calculation, typically jotted down on any available scrap of paper such as an [envelope](https://en.wikipedia.org/wiki/Envelope). It is more than a [guess](https://en.wikipedia.org/wiki/Guess) but less than an accurate [calculation](https://en.wikipedia.org/wiki/Calculation) or [mathematical proof](https://en.wikipedia.org/wiki/Mathematical_proof) [↑](#footnote-ref-1)
2. **Logical effort**: the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current [↑](#footnote-ref-2)
3. Ex) STFT=Temp: -125+VDD=1.1V +Fast NMOS+ Typical PMOS. [↑](#footnote-ref-3)