

CMOS 표준 셀의 자동설계

(An Automated Design of CMOS Standard Cells)

金 漢 興*, 慶 宗 旻**

(Han Heung Kim and Chong Min Kyung)

要 約

논리표현식과 설계규칙으로부터 CIF mask data를 자동적으로 생성하는 CMOS 표준 셀의 자동설계 방법을 제안하였다. N-well, P-well 혹은 twin-well 공정중의 선택과 guard band의 포함여부를 user가 지시할 수 있게 하였다. 또한, P-FET의 channel 폭을 변화시켜서 각 셀의 noise margin을 증가시키도록 하였다.

Abstract

We present an automated CMOS standard cell design methodology which generates a mask description in the CIF (Caltech Intermediate Form) from a user-given logic description and design rule. The resultant layout reflects the user's choice among N-well, P-well and twin-well process and user's decision whether the guard band is to be included or not. Noise margin of each cell was improved by carefully adjusting the channel width of P-FET.

I. 서 론

근래에 집적회로의 집적도가 VLSI 수준에 이룸에 따라 설계자동화를 통하여 칩설계에 소모되는 시간을 줄이려는 노력의 일환으로 빠른 turn-around time 과 적은 면적의 칩 실현을 가능케하려는 semi-custom 설계방식이 있는데, 대표적인 방식으로는 gate array와 표준 셀 방식이 있다. Gate array 방식은 한정된 logic macro를 가지고 있으며 배선단계만을 customize 함으로써 turn-around time은 매우 작지만, 면적소모가 큰편이다. 표준 셀 방식은 gate array에 비하여 turn-around time은 길지만 풍부한 표준 셀들을 이용

함으로써 적은 면적에 다양한 기능의 회로를 얻을 수 있다. 한편, full custom 설계 방식에 비하여 표준 셀 방식은 면적 소모는 많으나, 실제시간이 훨씬 작기 때문에 비교적 디지털 칩 설계에 널리 사용되고 있다. 종래의 표준 셀 방식에서는, 하나의 표준 셀을 설계하고 document하는데 1~2주일의 시간이 걸리며, 회로설계에 일부 변경 사항이 있거나 design rule이 변하는 경우에는 cell library에 있는 모든 셀 또는 그 일부를 변경해야 한다는 단점이 있다.^{1,2} 이러한 셀 생산성의 문제점을 보완하기 위하여, 이 논문에서는 설계규칙 및 프로세스 변수와 여러가지 CMOS 프로세스(N-well, P-well 및 Twin-tub process)의 선택을 파라메타로 하여 CMOS 표준 셀의 layout을 자동적으로 만들어 내는 프로그램 CELLGEN을 소개하려 한다.

II. CELLGEN의 입출력 File

CELLGEN에 의해 생성되는 표준 셀은 다음의 조건을 갖추고 있다. 첫째, 셀의 높이는 일정하고, 둘째, 입출력 단자들은 셀의 위나 아래에, 혹은 위 아래 모두에 있으며, 셋째, 입출력 단자들의 위치는 단위 길이의 정수배가 되도록 한다. CELLGEN의 입력 file로는

*正會員, 現代電子産業(株)半導體二事業本部
(Semiconductor Div. II Hyundai Electronics
Industries Co., Ltd.)

**正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)
接受日字: 1986年 3月 28日
(*본 연구는 부분적으로 한국과학재단의 재정적 지원에 의한 것임)

프로세스 변수에 따른 설계규칙의 변화를 쉽게 update 할 수 있는 layout을 얻기 위해 설계규칙 file을 설정한다. 이 file은 7-mask standard CMOS process의 각 mask data가 지켜야 하는 자체규칙, 포괄규칙 및 여유규칙 등을 포함한다. 이 규칙들로부터 각 mask의 size 및 그들간의 위치가 결정된다. 또한 N-well, P-well 및 Twin-tub CMOS 공정의 세가지 중에서 원하는 공정을 선택하는 변수 pr (1:N-well 공정, 2:P-well 공정, 3:Twin-tub 공정)이 있으며, latch-up을 방지하기 위한 guard ring을 만들것인지의 여부를 묻는 변수 gb (gb>0이면 guard ring을 포함하고 gb=0이면 안한다)가 있다.

CELLGEN은 셀에 대한 layout을 CIF 형태로 출력하는 외에 표준 셀 배치, 배선 시스템에서 필요한 다음과 같은 정보를 출력한다.

- 1) 셀의 폭 및 길이
- 2) 셀의 입출력 단자들의 이름 및 좌측으로부터의 순서
- 3) 좌측 첫번째 입출력 단자의 중심점의 X좌표
- 4) 입출력 단자간의 거리
- 5) 논리적으로 동등한 입출력 단자들의 집합

III. 셀 생성과정

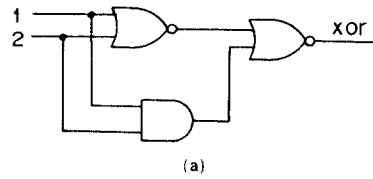
CELLGEN에서의 셀 생성과정은 입력해석, 기능블럭의 생성과 최종 셀의 생성 및 출력의 3단계로 크게 구성된다.

1. 입력해석

입력은 원하는 셀의 논리함수로 주어지며, 여기에서는 이 논리회로를 AOI(And-Or-Inverter), transfer gate 및 inverter 등의 기능블럭 단위로 분할한다. 예를 들어, 그림 1 (a)와 같은 XOR gate를 생성하고자 하는 경우에 CELLGEN에의 입력형태는 그림 1 (b) 와 같다.

이와 같은 입력형태는 다음과 같은 간단한 syntax를 사용하여 작성된다.

- 1) 논리함수의 이름과 이에 대한 함수식 사이에는 "="이 있다.
- 2) 각 블럭은 inv(inverter), t(transfer gate), nand(NAND gate) 등의 기호로 표시되며, 각 기능블럭의 입력 node 번호는 뒤에 따르는 괄호안에 나열된다.
- 3) 각 입력은 ",", "를 사용하여 구분한다.
- 4) 셀의 prime input은 1에서부터 시작하는 아라비아 숫자를 사용하여 표기한다.
- 5) 입력이 hard-wired된 경우에는 ",", " 대신 "|" 을 사용하여 구분한다.

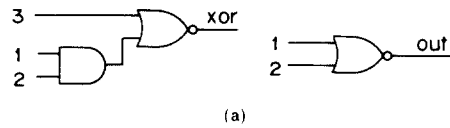


$$\text{xor} = \text{nor}(\text{nor}(1, 2), \text{and}(1, 2))$$

(b)

그림 1. XOR 게이트 셀에 대한 (a)logical diagram과 (b) CELLGEN에서의 입력형태

Fig. 1. (a) Logic diagram and (b) input format in CELLGEN for an XOR gate.



(a)

connect all 1's.
connect all 2's.
connect out and 3.

(b)

그림 2. XOR 게이트의 (a) 각 기능블럭 및 (b) 기능블럭간의 배선정보

Fig. 2. (a) Each functional block and (b) connectivity among functional blocks for an XOR gate.

그림 1 (b)의 입력형태로부터 XOR gate를 이루고 있는 기능블럭들을 추출하는 방법은 다음과 같다.

- 1) 셀을 이루고 있는 모든 gate를 출력 gate로부터 같은 거리에 있는 gate들의 group으로 분류한다.
- 2) 각 깊이에 있는 gate들을 살펴면서, 그 gate가 nand, nor, inverter 및 transfer gate인 경우에는 그 gate 및 그의 하부 깊이에 있고 그 gate와 상관되는 모든 and, or gate들을 묶어서 하나의 기능블럭으로 한다.
- 3) 이상에서 추출한 기능블럭들 간의 연결정보를 추출한다.
- 4) 이상의 결과를 소정의 형태로 출력한다.

이와같은 방법은 일종의 dynamic programming 기법을 사용하여 프로그램되었으며, XOR gate에 대한 결과는 그림 2와 같다.

그림 1 (a)의 logic diagram에서 출력 nor gate는 그 다음 깊이에 있는 and gate와 합하여 하나의 AOI 블럭을 형성하며, 나머지 nor gate는 그 자체로 하나의 AOI 블럭을 형성한다. 따라서 XOR gate는 그림2(a)

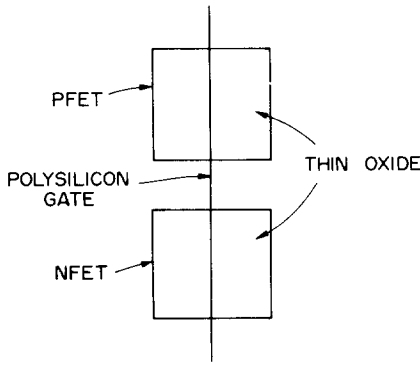


그림 3. Layout의 기본 단위
Fig. 3. Layout primitive.

에 나타난 바와 같은 기능블럭들로 나뉘어지며 그들에 대한 연결정보는 그림 2 (b)와 같이 된다.

2. 기능블럭 생성

여기에서는 그래프 이론의 결과를 도입하여 일반적인 AOI 기능블럭의 layout을 최소 면적으로 실현하는 방법을 설명한다. 기능블럭의 layout은 그림 3과 같은 layout primitive에 근거하는 방법을 사용하였다. 이와 같이 하여 기능블럭을 생성하는 방법은 다음과 같이 크게 3과정으로 나누어 진다. 첫째, 기능블럭의 각 prime input에 1:1 대응되는 각 layout primitive의 순서를 cell의 면적이 최소가 되도록 결정한다. 둘째, 그래프 모델로부터 기능블럭 안의 배선정보를 추출한다. 셋째, contact의 위치를 결정하고 기능블럭 안의 배선을 완료한다. 각 과정의 알고리즘을 설명하면 다음과 같다.

1) Layout primitive의 순서 결정

그림 4 (a)와 같은 논리 게이트의 트랜지스터 회로도는 그림 4 (b)와 같고 각각 N채널과 P채널 트랜지스터 간의 연결상태를 그래프로 표시한 것이 그림 4 (c)이다. 그림 4 (c)에서, Euler path가 존재하면 각 layout primitive 사이의 소오스나 드레인의 확산 영역을 공통으로 함으로써 모든 layout primitive의 thin oxide 영역을 접속 연결할 수 있다.¹²⁾ 그림 4 (d)는 Euler path를 고려하지 않고 layout한 결과인데, 같은 회로를 그림 5 (b)처럼 변형하여 그림 5 (c)의 그래프 모델에서 서로 dual 관계인 P-그래프와 N-그래프에서 공히 같은 순서의 2-3-1-4-5라는 Euler path를 찾아서 이 순서대로 layout한 결과가 그림 5 (d)로써 그림 4 (d)에 비해 적은 면적을 차지함을 알 수 있다.

본 논문에서는 Uehara 등이 제안한 algorithm을 채택하여 layout primitive의 순서를 결정토록 하였다.¹³⁾

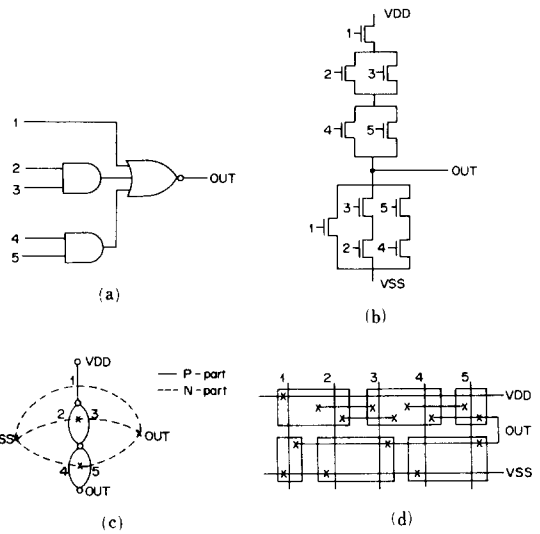


그림 4. 최적화되지 않은 셀 설계 예

- (a) 논리도, (b) 트랜지스터 회로도
- (c) 그래프 모델, (d) symbolic layout

Fig. 4. An example for a nonoptimized cell layout; (a) logic diagram (b) transistor-level circuit (c) graph model and (d) symbolic layout.

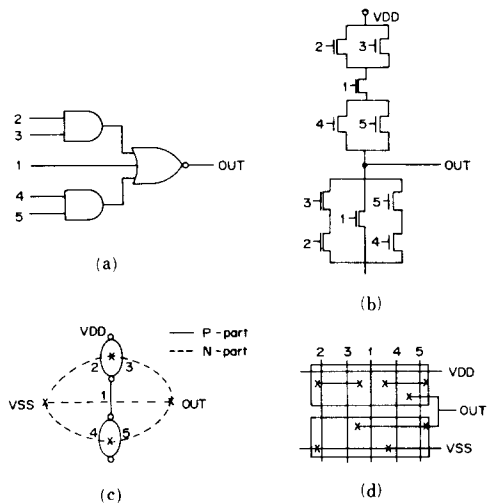


그림 5. 그림 4의 셀을 최적화한 경우

- (a) 논리도, (b) 트랜지스터 회로도
- (c) 그래프 모델, (d) symbolic layout

Fig. 5. An example showing an optimized layout; (a) logic diagram (b) transistor-level circuit (c) graph model and (d) symbolic layout.

2) 배선정보 추출

Euler path에 의해 prime 입력의 순서가 결정되면

그림 5 (c)의 그래프에서 각 vertex에 연결된 트랜지스터들로 구성된 netlist를 찾아내게 된다. 이 과정을 배선정보 추출이라하며 이에 대한 알고리즘을 설명하기로 한다.

i) 그림 6 (a)의 논리회로의 입력단에서부터 1번, 2번 논리 게이트에 대한 직·병렬 그래프 모델을 형성하면 각각 (b), (c)와 같이 된다.

ii) 그래프에서 각 vertex에 연결된 모든 edge에 대하여(각 edge의 번호가 i 일때 표 1의 소수 table에 의거한) i 번째 소수를 모두 곱한 값을 할당하면 (b), (c)

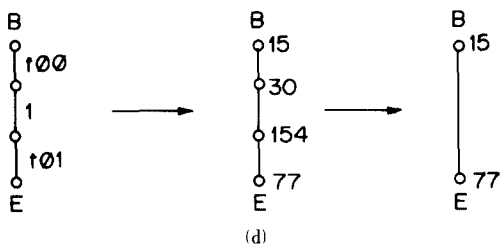
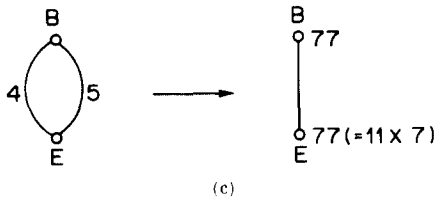
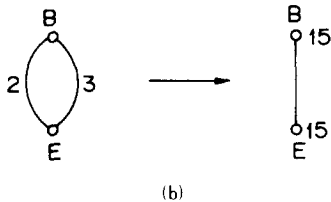
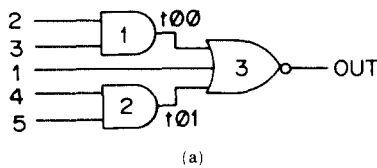


그림 6. P채널 트랜지스터간의 배선정보 추출과정
(a) 논리도, (b) 1번 게이트에 대한 그래프의 병렬 축약
(c) 2번 게이트에 대한 그래프의 병렬 축약
(d) 3번 게이트에 대한 그래프의 직렬 축약

Fig. 6. A procedure for extracting connectivity among P-channel transistors.
(a) logic diagram.
(b) parallel graphical compaction of 1-gate.
(c) parallel graphical compaction of 2-gate and
(d) series graphical compaction of 3-gate.

에서와 같이 병렬 그래프가 단일 edge로 축약된다.

iii) 다음에는 그림 6 (a)의 3번 게이트의 입력인 $t\phi\phi$, 1, $t\phi 1$ 에 대한 직렬 그래프를 축약하는데, 이 경우 (b)와 (c)에서 축약된 그래프를 그림 6 (d)에서 다시 직렬 축약되는 과정을 보이고 있다. 결과적으로 각각 vertex에 15, 30, 154, 77의 값이 할당된 직렬 그래프가 만들어지고, B-node에는 15, E-node에는 77의 값이 남고 중간 node값인 30과 154는 따로 출력된다.

이와 같이 직렬 축약되는 경우에는 B, E를 제외한 vertex의 배선정보가 출력되며, 병렬 축약되는 경우에는 배선정보가 B, E vertex에 남게된다. B, E vertex를 설정하는 이유는 i)에서 결정된 Euler path를 따르기 위함이다.

iv) 이와 같은 과정을 출력 게이트까지 행한다. 즉, 표 2와 같이 각 node의 출력된 수를 소수분해하여 netlist를 만들어냄으로써 전체적인 그래프 모델을 형성하는 번거모음을 피하면서 배선정보를 추출할 수 있다. N-FET 부분은 그래프의 duality를 사용하여 같은 방법으로 추출한다. 표 2에는 netlist에 속한 각 입력핀의 layout에서의 좌측으로부터의 순서도 표시했다.

3) Contact 위치 결정

이제 남은 일은 앞에서 얻은 배선 정보에 따라 contact의 위치를 결정하고 배선하는 일이다. 이 논문에서는 AOI 기능 블록내의 배선에는 N채널의 경우 3개

표 1. 소수 table
Table 1. Prime number table.

i	i번째 소수
1	2
2	3
3	5
4	7
5	10
.	.
.	.
.	.

표 2. 그림 6에서 얻어낸 배선정보(P-FET부분)
Table 2. Netlist extracted from Fig.6(P-FET part).

Node번호	각 Node의 소수분해	Netlist	Layout상의 각 입력핀의 순서
0	$30=5 \times 3 \times 2 \rightarrow 3 \times 2 \times 1$	3, 2, 1	2, 1, 3
1	$154=11 \times 7 \times 2 \rightarrow 5 \times 4 \times 1$	5, 4, 1	5, 4, 3
2(B-node)	$15=5 \times 3 \rightarrow 3 \times 2(XB)$	3, 2, VDD	2, 1, VDD
3(E-node)	$77=11 \times 7 \rightarrow 5 \times 4(XE)$	5, 4, Out	5, 4, Out

의 배선트랙(VDD 또는 VSS와의 배선을 위한 트랙과, 나머지 net들을 서로 엮갈리게 배선하기 위한 2개의 트랙을 할당하였으며 부록의 결과에 따라 P채널 트랜지스터의 width가 N채널의 그것보다 훨씬 커서 P채널의 경우에는 배선트랙의 수가 훨씬 많이 할당될 수 있다. Left edge algorithm⁽⁶⁾을 사용하여 net를 트랙에 할당한 후에 다음과 같은 방법으로 contact의 위치를 결정한다.

- i) Netlist를 분할한다. netlist의 숫자가 다음 숫자와 연속이 아니거나(즉, 다음 숫자보다 1이상 클때) 확산 영역이 분리된 곳에서 분할한다. 예를 들어 어느 netlist가 5, 4, 2, 1일때, 5번과 4번 사이에서 확산 영역이 분리된다면, 이 net를 분할한 결과는 {5} {4} {2, 1}의 세가지 contact set로 된다.
- ii) Contact set의 원소수가 홀수개이나, 짝수개이나에 따라서 그림 7과 같이 contact의 위치를 결정한다.

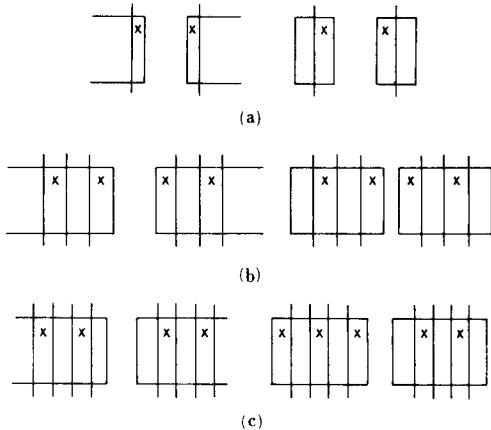


그림 7. 여러가지 경우의 contact의 위치;contact의 수가 (a) 1개인 경우, (b) 홀수(3개)인 경우, (c) 짝수(4개)인 경우

Fig. 7. Locating contact position for various cases;when the number of contacts is (a) one (b) odd(=3) and (c) even(=4).

3. 셀 생성

2절에서 생성된 기능 블럭들을 모아서 셀을 생성한다. 이는 CIF의 Call(C) 및 Transformation(T)를 사용하여 행해진다.

먼저 각 기능 블럭의 CIF data를 Call하여 설정된 위치에 놓는다. 중요한 것은 각 기능블럭이 설계규칙에 어긋남이 없도록 해야 한다는 것이다. 다음에는 1절에서 얻은 기능블럭들간의 배선정보에 따라 배선을 행함으로써 셀 생성이 완료된다. 이와같은 방법으로 생성된 몇가지 셀들이 그림 8과 그림 9에 나타나 있

며, symbolic layout 및 physical layout에서 각 layer를 나타내는 선들은 표 3과 같다. 그림 8과 그림 9의 symbolic layout에서 살펴보면, 기능블럭들간의 배선을 위해 외부배선 트랙이 있어야 함을 알 수 있다. 그러나 physical layout에서는 이들 외부배선 트랙들이 다음과 같은 방법으로 셀 내부에 포함되게 된다.

- i) P-FET 및 N-FET 사이에 하나의 배선트랙을 허용할 수 있는 공간이 존재할 수 있는가를 살펴본다.
- ii) 각 기능블럭의 P-FET 및 N-FET 부분에는 3개의 내부 배선트랙이 할당되므로 이들 배선트랙 중에서 내부배선시에 소용되지 않은 것이 있으면 cell 간의 배선용 트랙으로 사용한다.
- iii) 부록의 결과로 나타나는, P-FET 부분의 여분

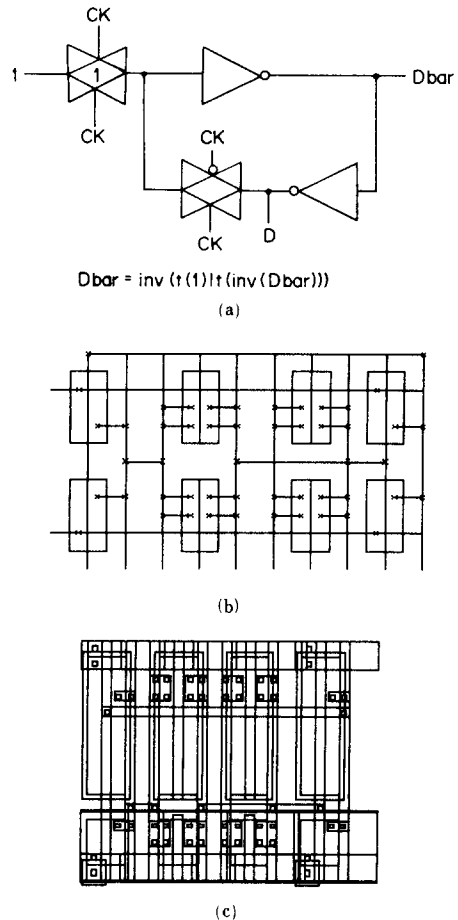
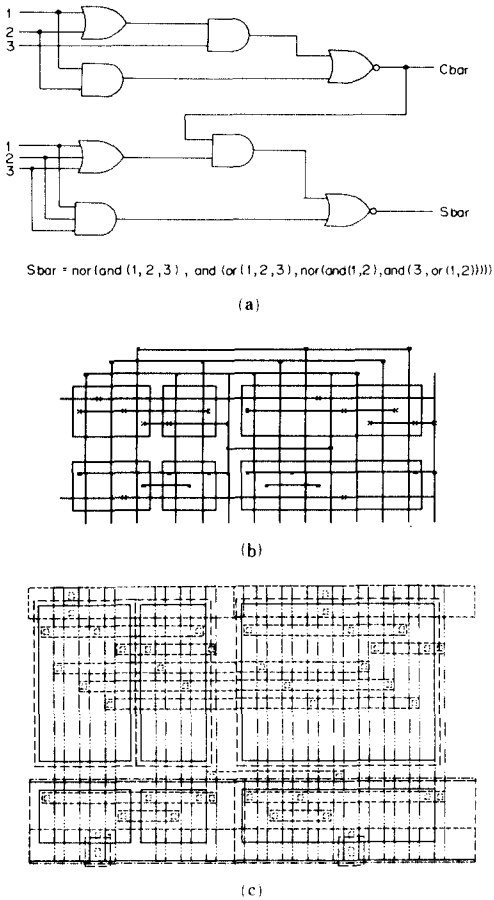


그림 8. Dynamic D-latch에 대한 (a) 논리도, (b) symbolic layout과 (c) P-well CMOS layout (pr=2)
 Fig. 8. (a) logic diagram (b) symbolic layout and (c) P-well CMOS layout (pr=2) for a dynamic D-latch.



$$Sbar = nor(and(1, 2, 3), and(or(1, 2, 3), nor(and(1, 2), and(3, or(1, 2)))))$$

그림 9. Full adder에 대한

- (a) 논리도, (b) symbolic layout과
- (c) P-well CMOS layout (pr=2)

Fig. 9. (a) logic diagram (b) symbolic layout and (c) P-well CMOS layout (pr=2) for a full adder.

표 3. Symbolic layout에서 각 layer를 나타내는 선들

Table 3. Line representation for each layer in the symbolic layout.

Diffusion layer	-----
Metal layer	-----
Poly layer
Contact	×

의 트랙을 외부배선에 사용한다.

iv) i)~iii)에서 조사한 트랙들로도 기능블럭들간의 배선을 모두 할 수 없는 경우에는 셀 생성을 포기하고 외부연결 사항으로 처리한다.

IV. 결론 및 고찰

본 논문의 결과로 나타나는 layout의 면적증가는 전문가의 수동 layout에 비하여 10% 이상인 것으로 나타났다. 각 기능블럭을 생성하는 과정에서는 본 논문의 결과보다 면적감소를 별로 기대할 수 없으나, 기능블럭들을 모아서 표준 셀을 생성할 때에는 셀 사이에 일정한 공간이 필요하기 때문에 될 수록 적은 수의 기능블럭으로 셀을 구성하는 것이 좋을 것이다. 또한, P-FET의 width를 결정하는데는 부록에서 설명하였듯이 모든 셀들의 noise margin과 그림8, 9에서 외부배선 net들을 셀 내부에서 배선할 수 있는 공간을 제공해야 하는 점을 고려하여야 한다.

본 논문에서 사용된 입력형태는 제한된 셀들에 대하여만 유효하며, multi-fanout이나 multi-feedback을 갖는 셀을 묘사하기에는 불편하게 되어 있는데, 이 점을 보완하기 위하여 표준 셀의 입력형태에 일반적인 logic simulator의 입력 file처럼 각 net간의 연결정보를 병용하였다.

실질적으로, logic 설계로부터 CMOS 표준 셀들의 layout을 자동생성하는 한가지 방법을 제안하고 프로그램 하였다. 본 program의 입력은 Boolean식과 유사한 논리식으로, 간단한 syntax를 사용하여 쉽게 작성할 수 있도록 하였다. 그래프 이론의 결과를 도입하여 layout의 면적을 최소화 하였고, 각 셀의 noise margin을 고려하여 P-FET의 channel width를 가변시키는 layout을 얻도록 하였다. 또한, technology file을 설정하여, 설계된 셀의 layout이 설계규칙의 변화에 맞게 쉽게 생성되도록 하였다.

附 錄

AOI 기능블럭의 P-FET Channel Width 결정

AOI 기능블럭의 그래프 모델로부터 소정의 방법을 통하여 P-FET 및 N-FET의 평균적인 channel length를 구하고, AOI 기능블럭을 inverter로 간략화하여 문제를 해결하고자 하는 생각을 제시하였다. 문제의 주요점은 각 AOI 블럭의 noise margin을 고려하여 P-FET의 channel width를 결정하는 방법이다. Layout에서 N-FET의 channel width는, 앞에서도 말한 바와 같이 3개의 배선타랙을 허용할 수 있는 정도로 되어야 하며, 따라서 그 이상의 크기를 요하지 않는다면 그의 size는 고정된다. 또한 N-FET 및 P-FET의 channel length는 설계규칙을 만족하는 최소의 size를 갖도록 하였으므로 문제해결의 변수는 P-FET의 cha-

channel width가 된다. CMOS 표준 셀 inverter의 경우에 N-FET 및 P-FET의 channel length 및 channel width를 각각 L_p, L_n 및 W_p, W_n 이라하면

$$(W_p/L_p)/(W_n/L_n) \approx 3 \quad (1)$$

일때 최대의 noise margin (VDD의 50% 정도)을 갖는다. 따라서 그림10과 같이 N-FET 및 P-FET의 평균적인 channel length $L_{n, ave}$ 및 $L_{p, ave}$ 를 결정한다면, AOI 기능블럭은 그림10과 같이 inverter로 축소될 수 있다.

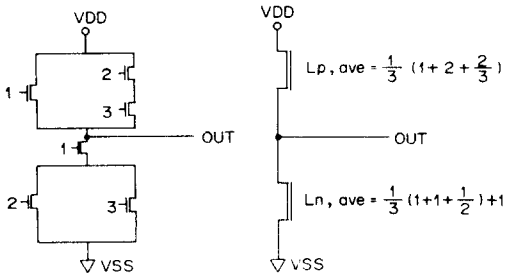


그림10. P-FET의 채널폭을 정하는 방식
 Fig. 10. A scheme for determining P-FET channel width.

따라서

$$(W_p/L_{p, ave})/(W_n/L_{n, ave}) \approx 3 \quad (2)$$

의 식을 사용하여 평균적으로 최대의 noise margin을 갖도록 W_p 를 결정할 수 있게 된다. 그러나 경우에 따라서 W_p 는 상당히 큰 값을 갖게 된다.(2-input NOR 게이트의 경우 $W_p=6W_n$ 이 된다) 이것은 일정한 높이를 가져야 하는 layout 조건상 불가능한 요구라 할 수 있다. 따라서 W_p 에 어떤 최대치를 설정할 필요가 있

게 된다. 그러므로 다음과 같은 방법을 사용하여 최종적으로 W_p 를 결정한다.

- i) (2)식에서 계산되는 W_p/W_n 이 어떤 일정한 값보다 크지 못하도록 W_p 의 한계값 $\max. W_p$ 를 설정한다.
- ii) W_p/W_n 이 $\max. W_p/W_n$ 보다 작을 경우에는 W_p/W_n 의 최대 정수 값을 택하여 W_p 를 결정한다.
- iii) 그렇지 않은 경우에는 $\max. W_p$ 값을 W_p 로 한다.

参 考 文 献

- [1] Nam Tosuntikool and Charles L. Saxe, "Automated design of standard Cells." *IEEE custom integrated circuits conf.* pp. 110-114, 1983.
- [2] M.W. Stebnisky, A. Feller, A.M. Smith, F. Borgini, S.S. Sharma, and M.J. McGinnis, "Approaches and tradeoffs in optimal Standard Cell Design." *ibid*, pp. 104-109, 1983.
- [3] Takao Uehara and William M. van Cleemput, "Optimal Layout of CMOS Functional Array," *IEEE Trans. on comp.*, vol. C-30, no. 5, pp. 305-312, May 1981.
- [4] Curtis T. McMullen and Ralph H.J.M. Otten, "Layout compilation of linear transistor arrays," *IEEE Proc. of ISCAS*, pp. 5-7, 1985.
- [5] 김한홍, CMOS 표준 셀의 자동생성, 한국과학기술원 석사학위 논문, 1986.
- [6] Guy Rabbat, *Hardware and Software Concepts in VLSI*, van Nastrand Reinbold, chap. 15, 1983.