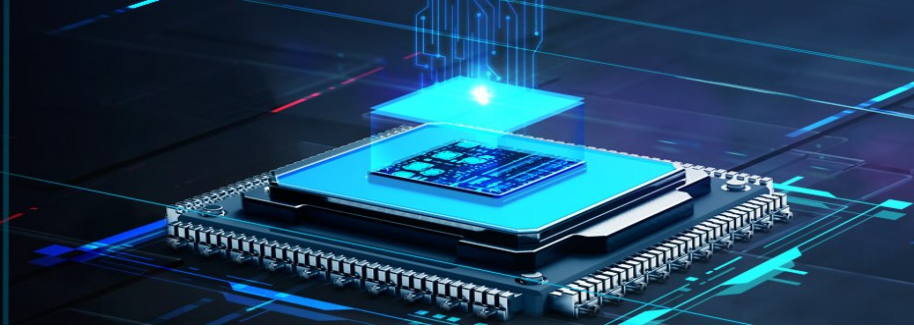


ETRI 0.5um CMOS Std-Cell DK:

표준 셀 D-플립플롭 설계

연구과제명	반도체 기술 개발 지원 고경력 전문인력 활용 사업(25JB1710)
연구기간	2025 년 6 월~2026 년 12 월
연구책임자	고상춘
기록자	국일호
확인자	
작성일자	2025 년 7 월 19 일



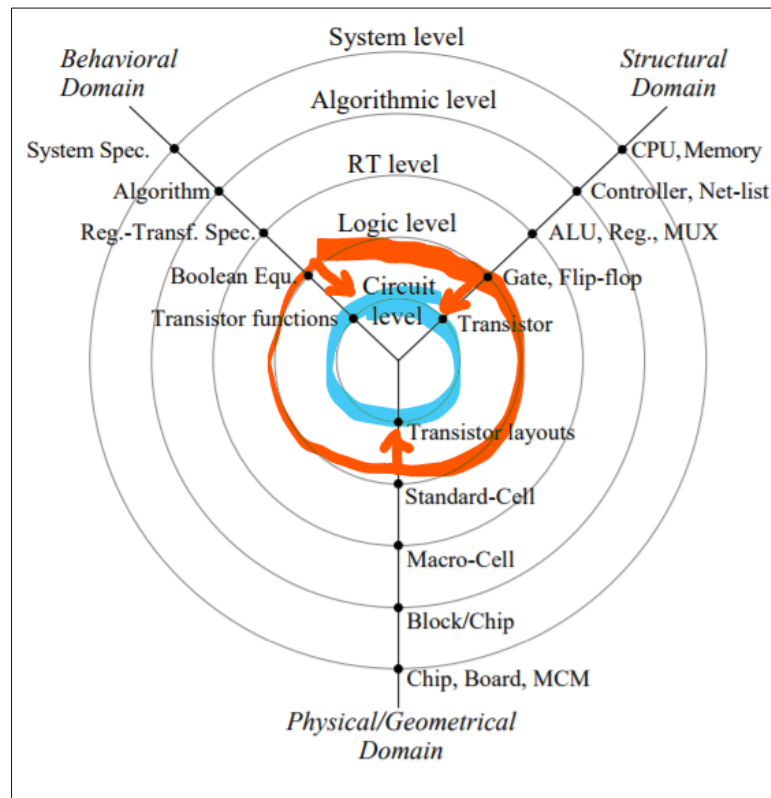
표준 셀 D-플립플롭 설계

목차

- I. 표준 셀(Standard Cell) 개요
 - II. 표준 셀(Standard Cell) 그리기 규칙
 - II-1. 표준 셀의 높이
 - II-2. 트랜지스터의 채널 크기
 - II-3. 금속 층의 활용
 - II-4. 금속 배선의 간격(Pitch or Tracks)
 - II-5. 비아와 포트
 - II-6. LEF: 자동 배선 규칙
 - III. 표준 셀 D-플립플롭의 회로 설계(XSchem)
 - III-1. D 플립 플롭의 기능 사양
 - III-2. 회로도 작성
 - III-3. 시뮬레이션
 - IV. 표준 셀 D-플립 플롭의 레이아웃 설계(Magic)
 - IV-1. 표준 셀 INVX1
 - IV-2. 표준 셀 NAND2X1
 - IV-3. 계층적 D 플립-플롭 레이아웃
 - IV-4. LVS
 - V. 표준 셀 D-플립 플롭 레이아웃 평가
-

I. 표준 셀(Standard Cell) 개요

표준 셀은 IC 설계에서 사용되는 반도체 설계 방법론(특히 디지털)의 일환이다. 모든 설계를 트랜지스터 수준에서 설계 하기에는 너무나 많은 수고가 든다. 이보다 한 단계 높은 논리회로 수준에서 설계하는 경우 자동화 도구의 조력을 받을 수 있다. 기본적으로 논리회로는 식(logic expression)으로 표현할 수 있고 수학적으로 알고리즘을 동원하여 최적화 할 수 있다. 뿐만 아니라 배치와 배선의 자동화(Place & Route Automation)를 꾀할 수 있다.



배선(routing)은 부품의 내부 설계에는 간여하지 않고 오직 금속 층을 사용하여 부품 사이의 기하학적 연결을 수행한다. 이때 사용될 부품은 자동 배선을 감안하여 크기와 핀의 간격과 같은 기준이 필요하다. 자동 배선 도구는 금속층을 활용하게 되는데 금속층의 두께(width)와 이격거리(separation) 그리고 비아(via)의 그리기 규칙(layout design rule)을 따른다.

II. 표준 셀(Standard Cell) 그리기 규칙

"내 칩(My Chip) MPW" 서비스[1]를 통해 제공하는 공정의 사용 가능한 금속층은 3개에 불과하며 금속의 두께와 비아의 크기 그리고 겹친 비아(stacked via) 금지등 제한 규정이 일반적이지 않다(크다!). 자동 배선 도구가 이 규정을 따라야 함은 물론이다. 자동화 배선 도구에서 사용 가능하도록 아래와 같이 셀 그리기 규칙을 수립하였다. 표준 셀 사이의 배선 자동화 규칙은 LEF(Library Exchange Format)[10]로 규정한다. "내 칩(My Chip) MPW" 공정용 표준셀 '디자인 킷'은 표준 셀과 자동 배치와 배선용 LEF를 제공한다[2].

[주] 표준 셀은 RTL 합성과 자동 배치 배선을 위한 것이다. 추상화 수준을 높일 목적이 아니라면 단지 레이아웃 만으로는 아무런 실용적 의미를 갖지 못한다. “내 칩 MPW” 오픈-소스 디자인 킷[2]은 합성용 리버티(Liberty)와 합성후 네트 시뮬레이션용 베릴로그 모델 그리고 자동배선용 LEF를 표준 셀과 함께 제공한다.

Liberty*: [khu_etri05_stdcells.lib](#)

Std-Cell Verilog Model: [khu_etri05_stdcells.v](#)

LEF: [etri050_stdcells.lef](#)

[*] 표준셀의 타이밍 자료는 무효다.

II-1. 표준 셀의 높이

셀의 높이는 자동 배선 도구의 성능에 영향을 받지만 이를 일률적으로 장담하기는 어렵다. 셀을 조밀하게 그릴 경우 자동 배선에 실패할 수 있고 넓게 그릴 경우 유휴 공간이 늘어난다. 일반적으로 표준 셀은 9 또는 12 트랙(Track)이다. ‘디자인 킷’[2]의 표준 셀은 몇가지 베릴로그 설계를 활용하여 자동 배치와 배선을 시도한 끝에 셀의 높이는 13(12)개 트랙이 지나는 것으로 확정했다.

II-2. 트랜지스터의 채널 크기

트랜지스터의 채널 크기는 PDK[1]의 최소 권장 보다 큰 폭 3um, 길이 0.6um 로 하였다.

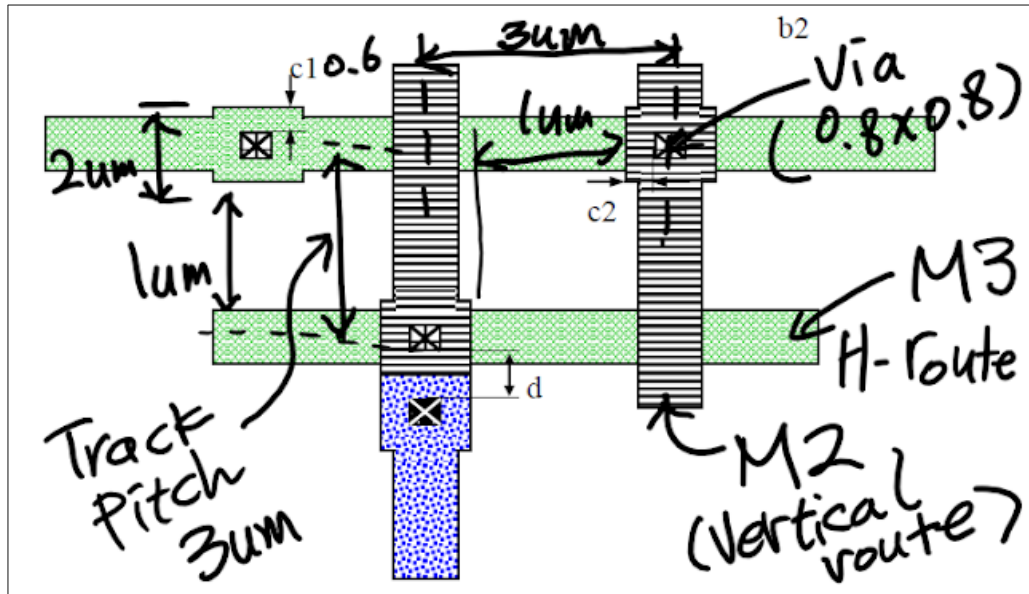
II-3. 금속 층의 활용

금속1 층(metal 1 layer)은 단위 셀 내의 지역 배선(cell internal route)에 사용한다. 자동 배선에서 금속1 층의 사용은 수평 배선에 사용할 수 있으나 극히 제한적이다. 금속 2와 3층(metal 1 and metal 3 layer)은 셀 사이의 전역 배선(inter-cell global route)에 활용한다. 금속 2층은 수직(horizontal route), 금속 3층은 수평 배선(vertical route)에 사용한다.

[주] 금속 2와 3층의 수평-수직 배선 활용은 LEF에서 변경 가능하다.

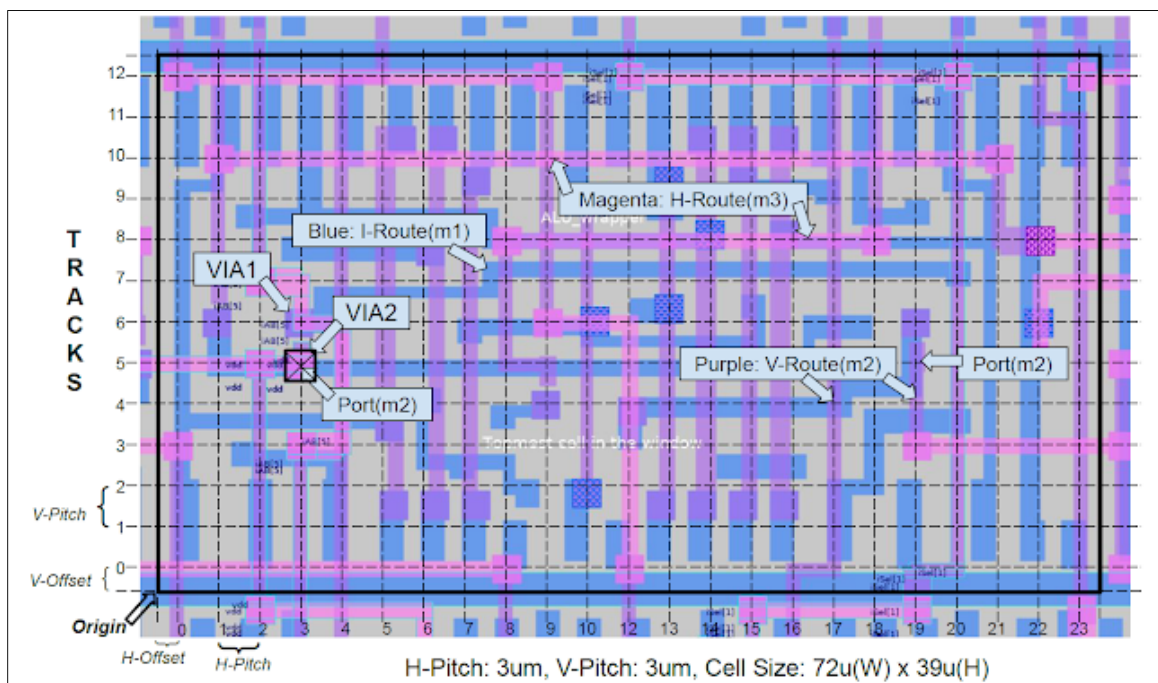
II-4. 금속 배선의 간격(Pitch or Tracks)

레이아웃 그리기 규칙의 배선용 금속 2와 3층의 두께, 비아의 크기를 감안하여 배선이 지나게 될 경로(track)의 간격은 수평과 수직 공히 3um 로 한다.

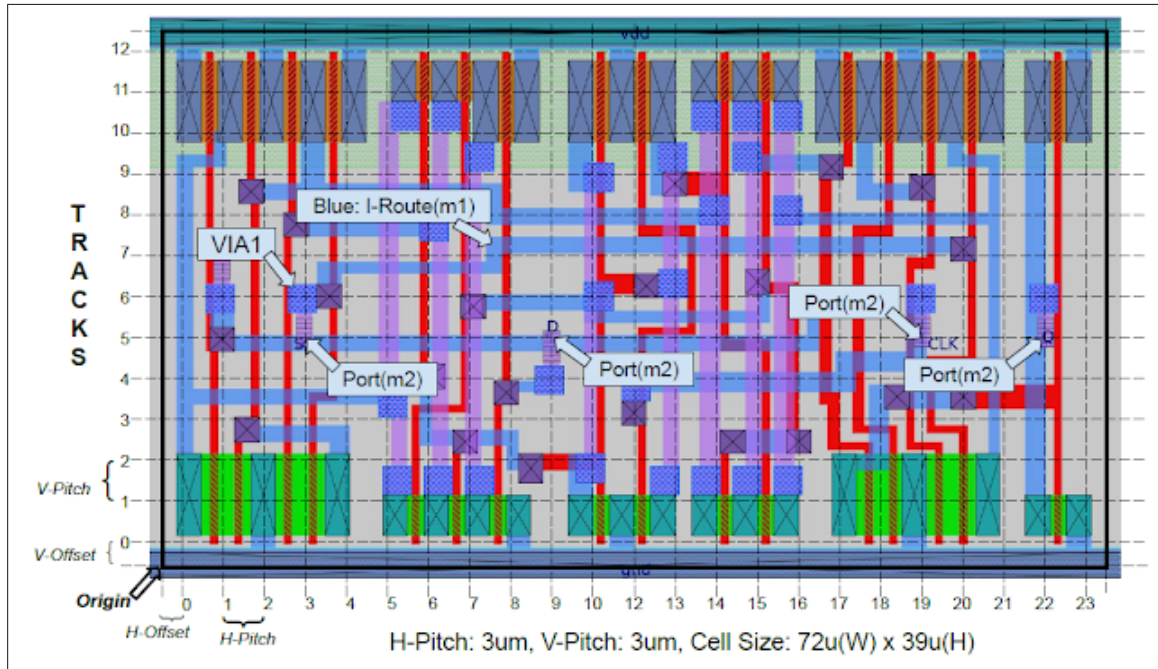


II-5. 비아와 포트

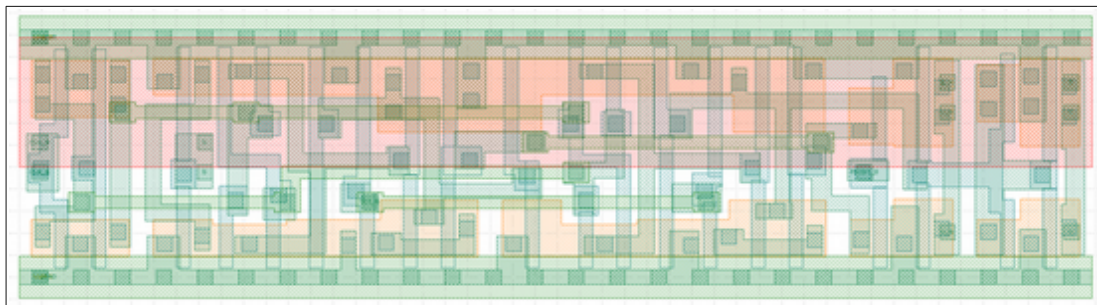
아래 그림은 실제 배선이 이뤄진 레이아웃의 일부 모습이다. 붉은 사각형은 표준 셀 중 복잡도가 높은 D 플립-플롭 이다. 셀 내부의 지역 배선에서 금속 1층(파란색)이 사용 되었고 그 위로 금속 2(보라색)의 수직 배선, 최상층 금속 3(분홍색) 배선이 트랙을 따라 지나고 있다. 큰 정사각형 분홍색은 수직과 수평 배선 사이의 비아2(via2), 보라색 정사각형은 금속1과 2 사이의 비아1(via1)이다.



비아1은 셀의 포트를 전역 배선으로 연결한다. 셀 내부의 배선은 가급적 금속1 만을 사용하지만 피치 못할 경우 금속 2를 최소한으로 사용한다. 이때 셀의 포트가 자동 배선에 막히지 않도록 수직 배선으로만 사용하는 것을 원칙으로 한다. 표준 셀 그리기 규칙을 적용한 D 플립-플롭 표준 셀의 모습은 아래와 같다. 폴리 실리콘 콘택과 디퓨전 콘택은 비교적 자유롭지만 비아1(via1)은 자동 배선 도구와 보조를 맞춰야 하므로 반드시 트랙 선상에 놓도록 한다.



많은 금속층을 가진 공정의 경우 전역 배선을 할 수 있는 여유가 많고 대부분 겹친 비아(stacked via)를 허용하므로 표준 셀을 조밀하게 그릴 수 있다. 5개 금속층을 가진 SkyWater130[3]의 D-FF(dffbbn1)의 내부 모습은 아래와 같다. 셀 내부가 매우 조밀한 것을 볼 수 있다.



II-6. LEF: 자동 배선 규칙

'내 칩 MPW 디자인 킷[2]'의 표준셀 [LEF\(etri050_stdcell.lef\)](#)에서 규정한 자동배선 규칙은 다음과 같다. 금속 층의 배선 방법과 비아 규칙(컷과 익스텐션)이 정의되었다. 이 규칙은 자동 배치배선 도구에서 활용된다.

```
#-----
# LEF file for Route & Via Rule
# Ported from osu050 by GoodKook@gmail.com
```

```

UNITS
  DATABASE MICRONS      1000 ;
END UNITS
USEMINSPACING      OBS ON ;
USEMINSPACING      PIN OFF ;
CLEARANCEMEASURE    EUCLIDEAN ;
MANUFACTURINGGRID    0.15 ;

.....
LAYER cc            # 콘택 컷 (구멍) 크기 규칙
  TYPE      CUT ;
  SPACING 0.9 ;
END cc
LAYER metall        # 금속1 층 규칙
  TYPE      ROUTING ;          # 자동 배선기
  DIRECTION HORIZONTAL ;      # 수평 배선용으로 사용가능
  PITCH     3.0 ;              # 배선 간격
  OFFSET    1.5 ;
  WIDTH     0.9 ;              # ETRI050 Rule: WIDTH=0.8
  SPACING   1.05 ;             # ETRI050 Rule: SPACING=1.0
  RESISTANCE RPERSQ 0.09 ;     # 저항 성분
  CAPACITANCE CPERSQDIST 3.2e-05 ; # 커패시턴스
END metall
LAYER via1          # 비아1
  TYPE      CUT ;
  SPACING   0.9 ;
END via1
LAYER metal2        # 금속2 층은 수직 배선용
  TYPE      ROUTING ;
  DIRECTION VERTICAL ;
  PITCH     3.0 ;
  OFFSET    1.5 ;
  WIDTH     1.05 ;             # ETRI050 Rule: WIDTH=1.0
  SPACING   1.05 ;             # ETRI050 Rule: SPACING=1.0
  RESISTANCE RPERSQ 0.09 ;
  CAPACITANCE CPERSQDIST 1.6e-05 ;
END metal2
LAYER via2          # 비아2
  TYPE      CUT ;
  SPACING   0.9 ;
END via2
LAYER metal3        # 금속3은 수평 배선용
  TYPE      ROUTING ;
  DIRECTION HORIZONTAL ;
  PITCH     3.0 ;
  OFFSET    1.5 ;
  WIDTH     1.2 ;              # ETRI050 Rule: WIDTH=1.2
  SPACING   1.05 ;             # ETRI050 Rule: SPACING=1.0
  RESISTANCE RPERSQ 0.05 ;

```

```

CAPACITANCE CPERSQDIST 1e-05 ;
END metal3
VIA M2_M1 DEFAULT      # 비아1의 규정
  LAYER metal1 ;        # 비아1 구멍을 감싸는 금속1은 중심에서 사방으로 팽창
  RECT -1.050 -1.050 1.050 1.050 ;
  LAYER via1 ;          # 구멍은 중심에서 사방으로 확장 규정
  RECT -0.450 -0.450 0.450 0.450 ;
  LAYER metal2 ;
  RECT -1.050 -1.050 1.050 1.050 ;
END M2_M1
VIA M3_M2 DEFAULT      # 비아2의 규정
  LAYER metal2 ;
  RECT -1.050 -1.050 1.050 1.050 ;
  LAYER via2 ;
  RECT -0.450 -0.450 0.450 0.450 ;
  LAYER metal3 ;
  RECT -1.050 -1.050 1.050 1.050 ;
END M3_M2

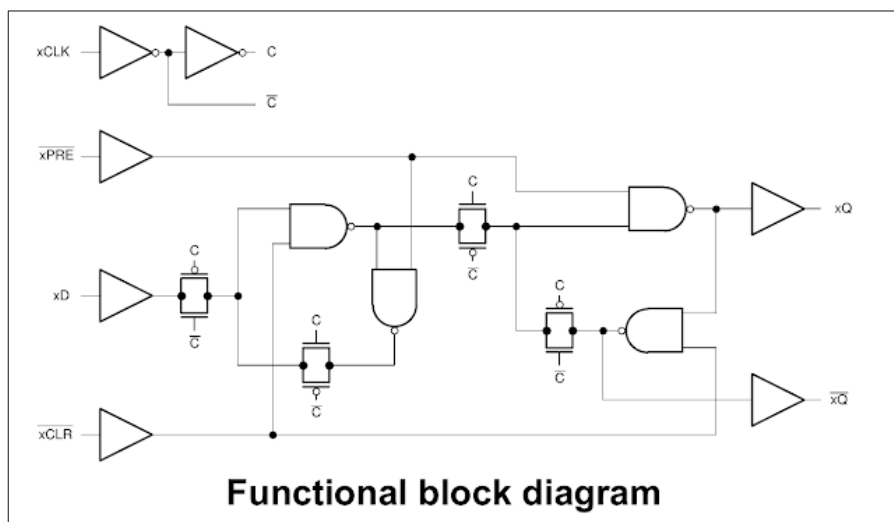
```

III. 표준 셀 D-플립플롭 회로 설계

플립 플롭은 디지털 정보(0 또는 1)의 저장소로서 순차 회로의 핵심이다[4]. 표준 논리회로를 활용하여 D 플립플롭을 설계한다.

III-1. D-플립플롭의 기능 사양

비동기 셋(async. set)과 리셋(async. re-set)을 가진 상승 엣지 트리거(positive-edge trigger) D-형 플립-플롭을 표준 셀로 설계하기로 한다. 잘 알려진 TTL 7474의 CMOS 버전 74HCT74의 내부 회로 구성을 차용하여 기능 사양(functional specification)으로 삼았다. D-플립플롭의 기능도는 아래와 같다.



[그림출처] D-Type Positive-Edge-Triggered Flip-Flops With Clear and Preset[5]

III-2. 회로도 작성(D-FF Schematic)

XSchem으로 기능 사양서에 따라 회로도를 아래와 같이 작성한다. 이번 실습은 회로도와 레이아웃 폴더를 분리하였다. 회로도 작성 실습 디렉토리 구조는 아래와 같다.

```
$ cd ~/ETRIO50_DesignKit/devel/Tutorials/1-4_StdCell_DFFSR/Sch
```

```
$ tree
```

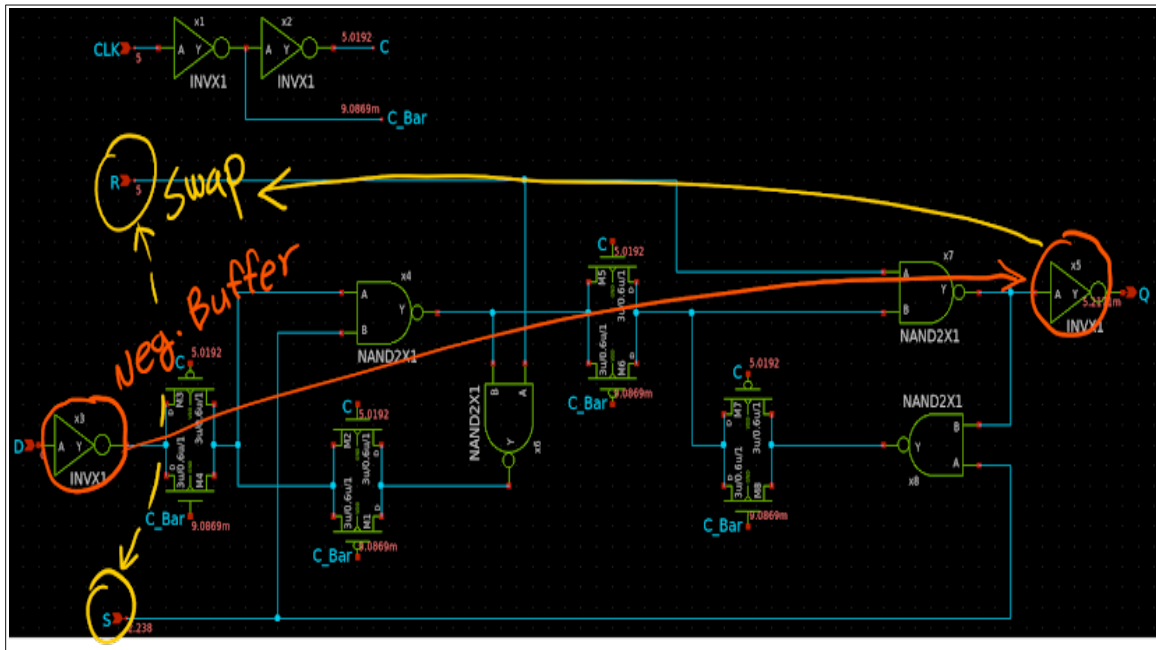
```
.
├── DFFSR_ALL.sch
├── DFFSR.sch
├── DFFSR.sch_Err
├── DFFSR.sym
├── DFFSR_TB.sch
├── INVX1.sch
├── INVX1.sym
├── INVX1_TB.sch
├── NAND2X1.sch
├── NAND2X1.sch_Err
├── NAND2X1.sym
├── NAND2X1_TB.sch
├── simulation
│   ├── DFFSR.spice
│   ├── DFFSR_TB.out
│   ├── DFFSR_TB.raw
│   ├── DFFSR_TB.spice
│   ├── INVX1_TB.out
│   ├── INVX1_TB.raw
│   ├── INVX1_TB.spice
│   ├── NAND2X1.spice
│   ├── NAND2X1_TB.out
│   ├── NAND2X1_TB.raw
│   └── NAND2X1_TB.spice
```

회로도 작업 폴더로 이동한 후 XSchem 실행.

```
$ xschem DFFSR.sch
```

[주] 편의를 위해 예제의 완성된 회로도가 작업 폴더에 모두 제공된다.

하위 회로로 NAND2X1과 INVX1를 별도로 작성하여 계층적 회로로 작성 하였다. 트랜지스터의 폭과 길이는 미리 만들어 둔 표준 셀과 동일하게 주었다.



플립-플롭에서 전자정보는 NAND 게이트의 궤환(feed-back)에 의해 유보된다. 이 궤환의 경로에 외부 영향을 배제 하기 위해 D 입력에 반전 버퍼(inverter)를 달았다. 아울러 출력의 궤환작용이 외부에 영향을 주지 않도록 역시 반전 버퍼가 있다. 입력과 출력에 모두 반전 버퍼를 달고 있지만 전역 신호 리셋 R (/CLEAR)과 셋 S (/PRE)은 반전을 상쇄하기 위해 맞바꿨다.

III-3. 시뮬레이션

트랜시언트 시뮬레이션(transient simulation) 결과는 아래와 같다. 비동기 R(/CLR)과 S(/PRE)의 동작과 클럭의 상승 엣지에 동기된 정보보유(latch)의 동작을 확인 하였다.

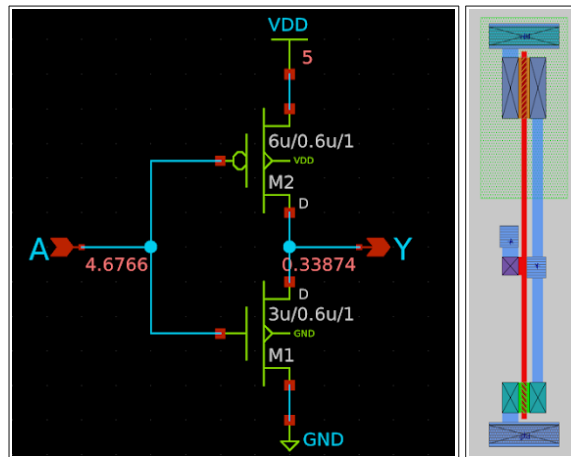


IV. 표준 셀 D-플립 플롭의 레이아웃

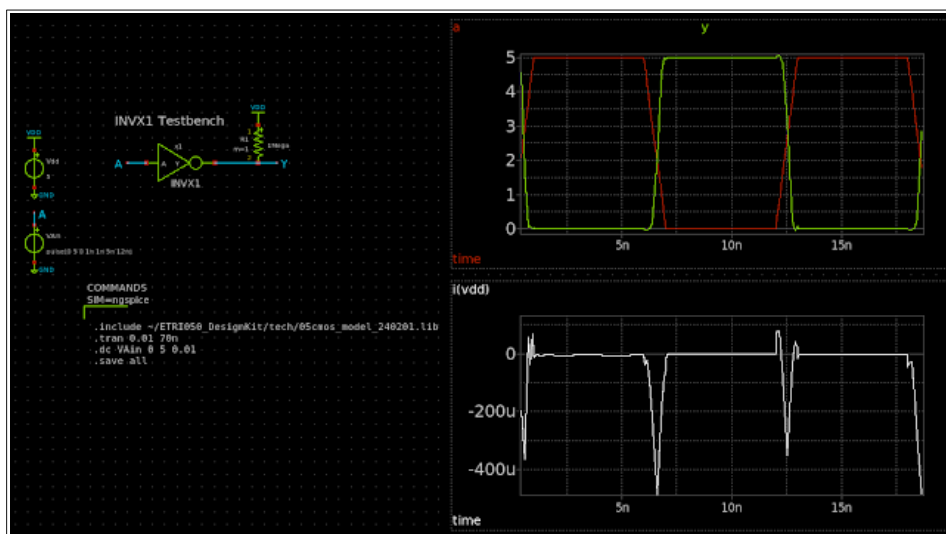
기능이 검증된 D 플립-플롭의 회로도를 사양으로 레이아웃을 그려보자. 회로도에 NAND 와 INV 게이트 그리고 스위치용 p/n-mos 쌍이 사용되었다. 레이아웃 에서도 이에 대응하는 표준 셀을 준비한다.

IV-1. 표준 셀 INVX1

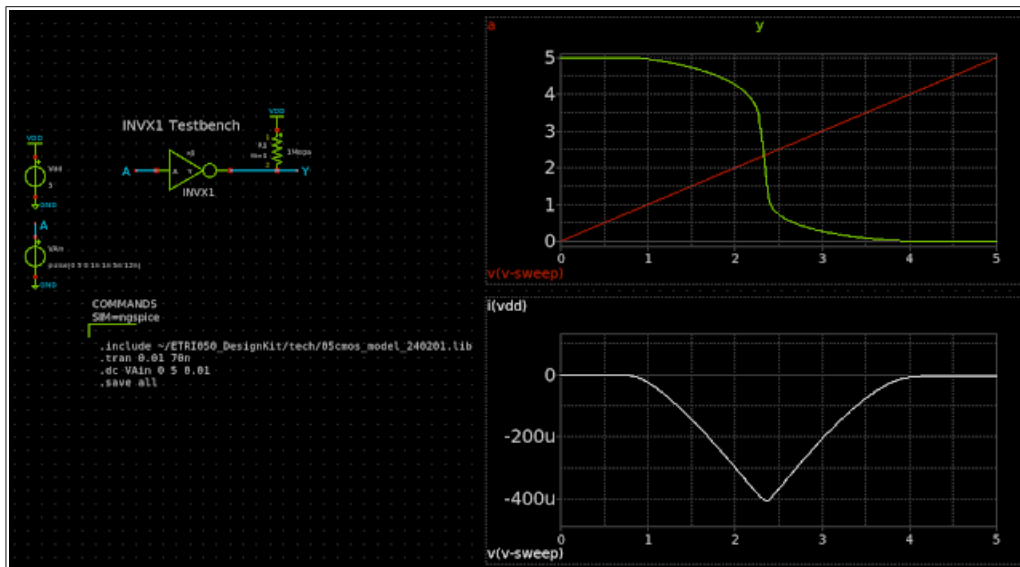
INVX1는 구동력(driving power)이 1인 인버터(inverter)다. n-mos의 채널(전자 이동한다)과 p-mos의 채널(홀이 이동한다)은 전하 이동도(mobility)가 2배 차이가 나는 것이 실험적으로 알려져 있다[6]. 이동도가 서로 다른 채널을 상보적(complementary)으로 사용하는 CMOS 에서 이를 해결하는 방법으로 p-mos 트랜지스터의 채널 폭을 두배로 늘린다. 집적도에서 손해보는 방법이지만 가장 수월하다.



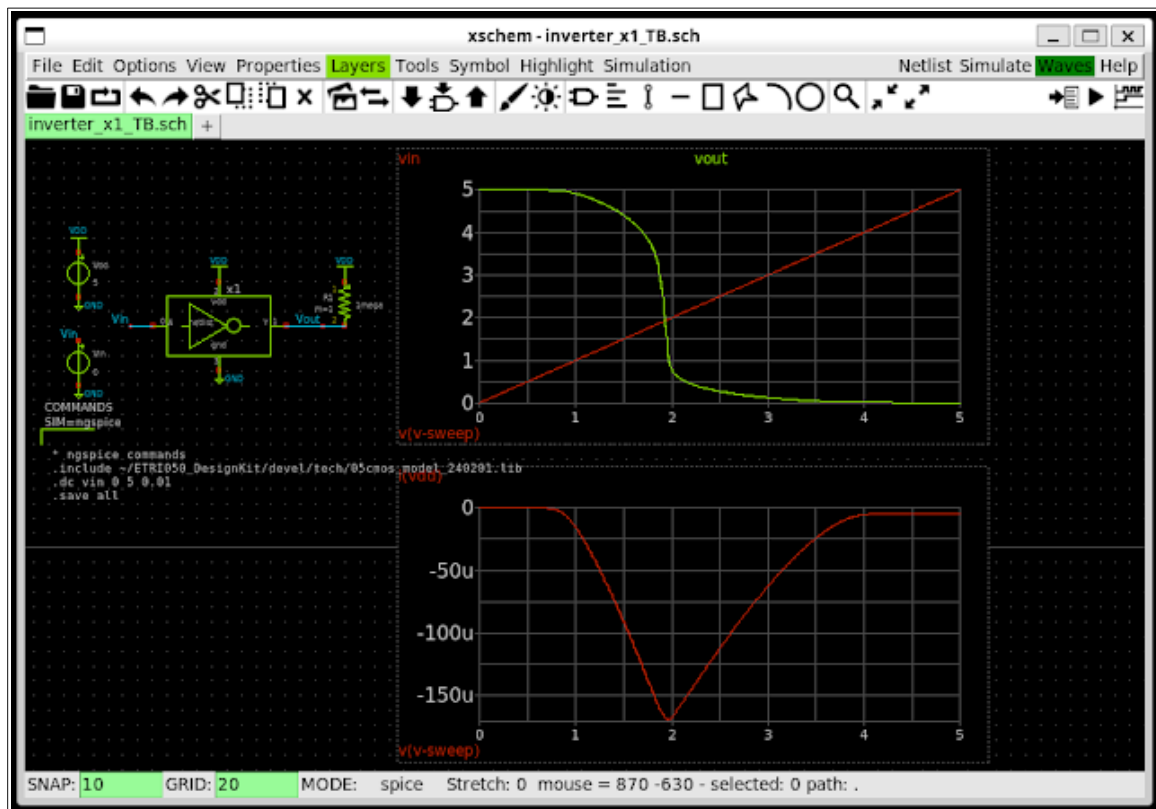
트랜시언트 시뮬레이션(Transient Simulation)은 수평축이 시간이다. 입력에 대하여 DUT가 반응하는 시간을 평가한다.



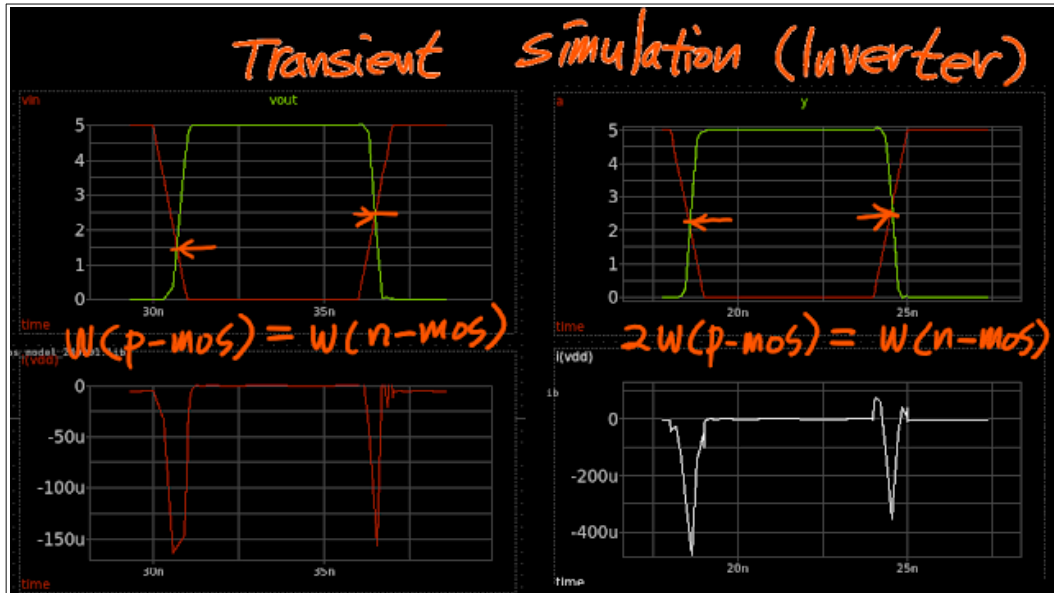
전압 증가 시뮬레이션(V-Sweep Simulation)은 수평축이 전압이다. 입력 전압에 대하여 DUT의 출력 전압을 평가한다.



앞의 실습에서 p와 n-mos의 채널 폭이 동일한 인버터의 입력 전압에 대한 출력의 반응(voltage sweep)과 비교해 보자. 반전 하강하는 출력 곡선이 매우 가파르다. n-mos가 빠르게 반응하기 때문이다.

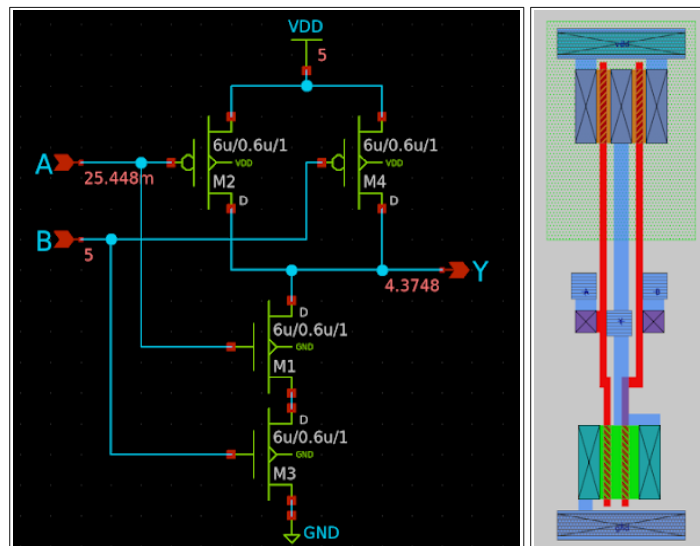


트랜지언트 시뮬레이션으로 비교해 보면 두 인버터 출력의 상승 반전과 하강 반전되는 시간의 차이가 드러난다. 좌측 그래프는 p-mos와 n-mos의 채널 폭이 동일한 경우다. 우측 그래프는 p-mos 채널 폭이 n-mos 채널 폭의 2배인 경우다.

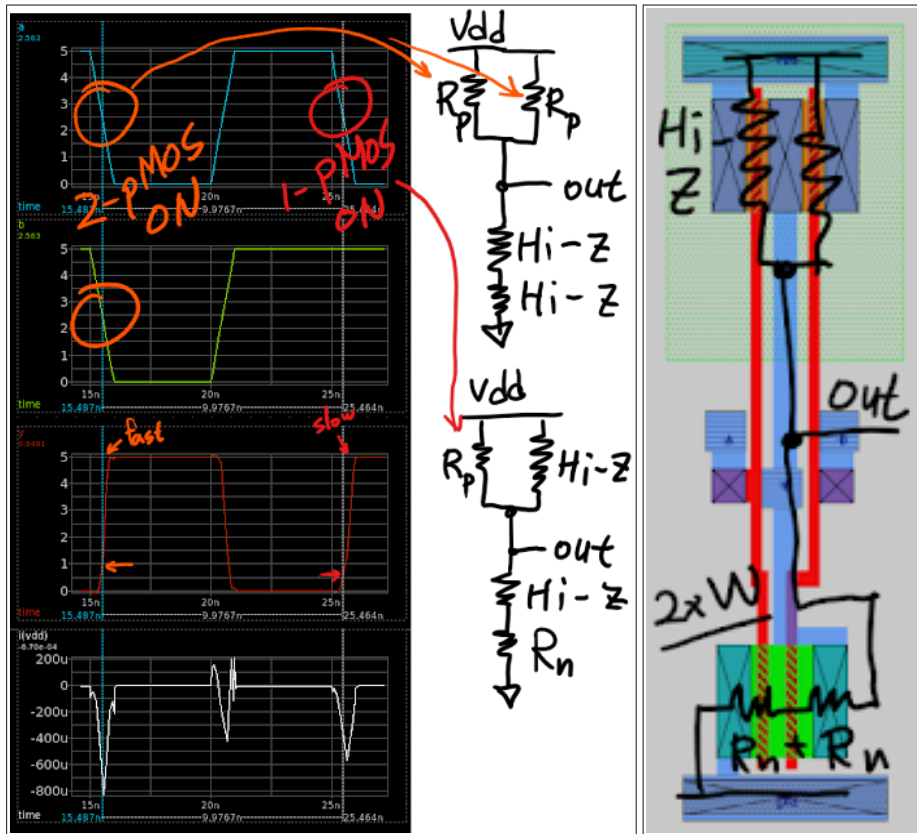


IV-2. 표준 셀 NAND2X1

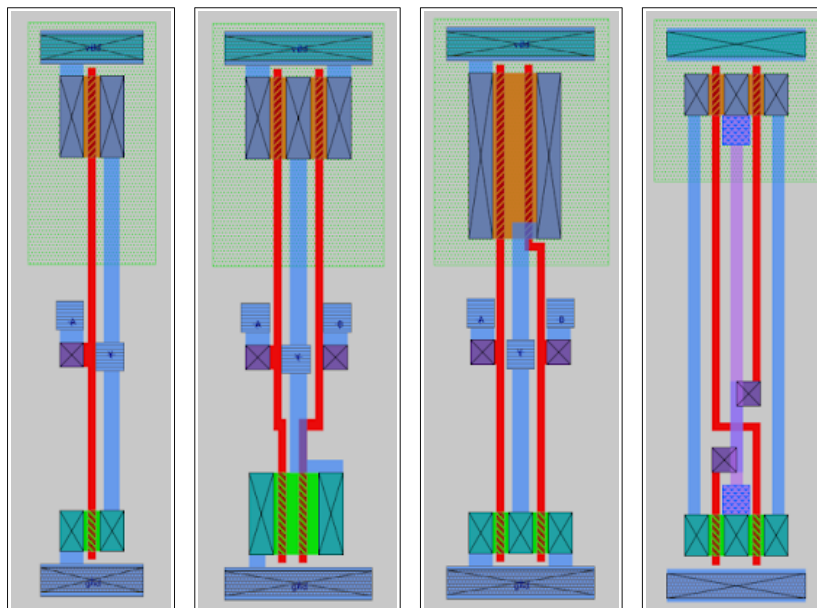
표준 셀 NAND 게이트의 회로도와 레이아웃은 아래 그림과 같다.



NAND2X1의 트랜지언트 시뮬레이션 결과는 아래와 같다. 두 입력이 모두 p-mos를 켜는 경우 vdd와 출력의 작아져서 반응 시간이 빠르다.



직렬 연결된 두 n-mos가 켜지는 경우 채널 길이(저항)가 두배가 된다. 이를 보상하기 위해 채널 폭을 늘렸다. 두 p-mos는 병렬 연결이다. 둘 중 하나만 켜지는 경우는 직렬연결된 두 n-mos와 균형을 이루지만 모두 켜지는 경우 빠르게 동작한다. 이를 보상하기 위해 n-mos의 채널 폭을 두배로 늘였다. 표준 셀 INVX1, NAND2X1, NOR2X1, SWITCH2X1의 레이아웃을 비교해 보면 트랜지스터의 채널 폭을 달리하는 이유를 같은 논리로 설명할 수 있다.

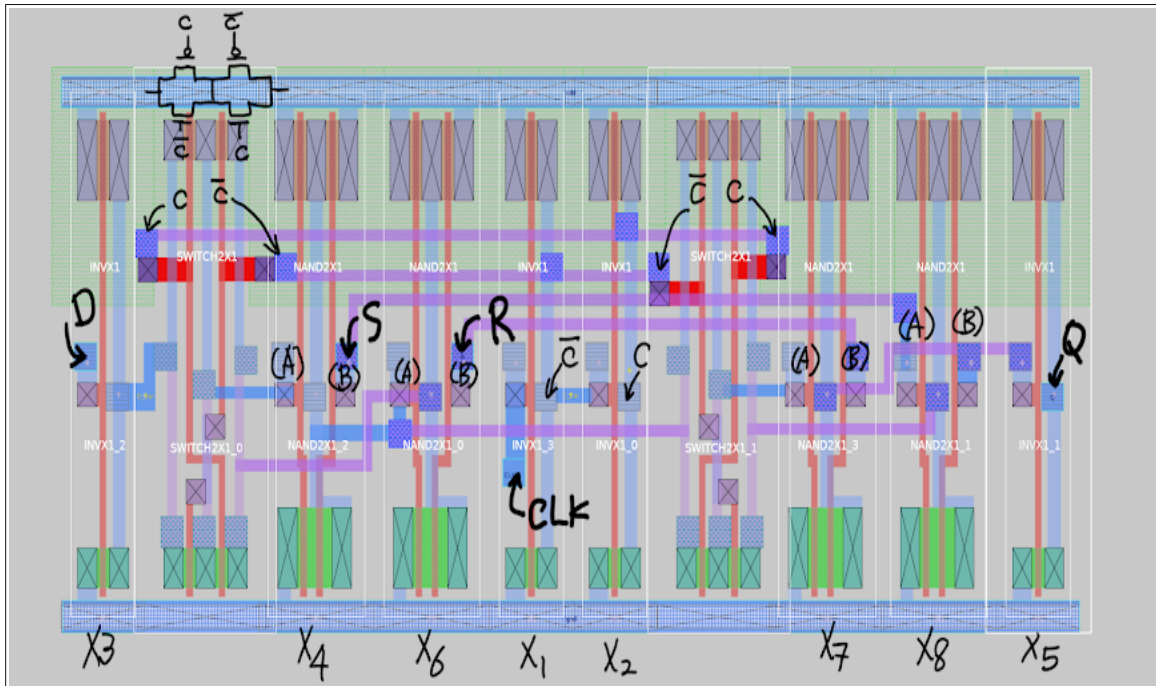


IV-3. 계층적 D 플립-플롭 레이아웃

미리 구축해 놓은 논리 게이트들의 표준 셀 레이아웃을 활용하여 회로도와 동일한 D-플립 플롭을 그린다.

```
$ cd ~/ETRI050_DesignKit/devel/Tutorials/1-4_StdCell_DFFSR/Layout
$ magic -d XR DFFSR74
```

표준 게이트를 하위회로로 두고 있는 계층적 레이아웃이다. 레이아웃의 이름은 DFFSR74.mag 다.



회로도의 p와 n-mos 쌍의 스위치를 표준 셀로 만들어 하위회로로 활용 하였다.

IV-4. LVS

회로도와 레이아웃의 동일성 검증을 위해 LVS를 실시한다. Magic 레이아웃의 이름은 DFFSR74 이며 회로도의 D 플립-플롭의 이름은 DFFSR 이다. 회로도의 네트리스트는 시뮬레이션을 위해 미리 생성해둔 DFFSR_TB.spice 다. 이 네트리스트 내에 하위회로로 존재하는 DFFSR와 레이아웃에서 추출한 네트리스트의 하위회로 DFFSR74를 비교한다. LVS의 결과 보고서는 LVS_DFFSR74_DFFSR.txt 로 하였다.

```
$ netgen -noc -batch lvs \
  "/DFFSR74.spice DFFSR74" \
  "../Sch/simulation/DFFSR_TB.spice DFFSR" \
  ./netgen_setup.tcl LVS_DFFSR74_DFFSR.txt
```

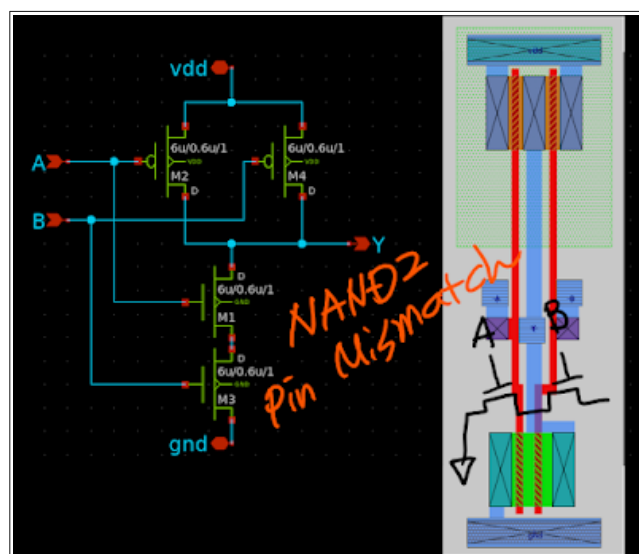
[주] 설계는 도구의 반복 실행이다. 매번 명령줄에서 긴 명령을 입력하려면 피곤하다. 스크립트를 작성해 두도록 한다. 위의 LVS 실행 스크립트는 run_lvs.sh 다.

```
$ ./run_lvs.sh DFFSR74 DFFSR
```

LVS 수행 후 결과는 두 회로의 네트리스트 불일치(mismatch)로 판명 되었다. 보고서를 열어 원인을 살펴보자. 처음 발견된 불일치는 NAND2X1의 핀 목록(pin list)에 불일치가 있다고 한다.

```
ome > goodkook > ETRI050_DesignKit > Tutorials > 1-4_StdCell_DFFSR > Layout > LVS_DFFSR74_DFFSR.txt
1
2 Subcircuit summary:
3 Circuit 1: NAND2X1                                |Circuit 2: NAND2X1
4 -----|-----
5 pfet (2)                                           |pfet (2)
6 nfet (2)                                           |nfet (2)
7 Number of devices: 4                             |Number of devices: 4
8 Number of nets: 6                                |Number of nets: 6
9 -----|-----
10 Netlists match uniquely.
11
12 Subcircuit pins:
13 Circuit 1: NAND2X1                                |Circuit 2: NAND2X1
14 -----|-----
15 vdd                                                |vdd
16 B                                                  |A **Mismatch**
17 A                                                  |B **Mismatch**
18 Y                                                  |Y
19 gnd                                                |gnd
20 -----|-----
21 Cell pin lists for NAND2X1 and NAND2X1 altered to match.
22 Device classes NAND2X1 and NAND2X1 are equivalent
```

Netgen은 NAND2X1의 레이아웃과 회로도의 두 입력 순서가 바뀐 것을 찾아내 이를 불일치 판정했다. NAND 게이트의 논리를 따져보면 두 입력의 순서가 바뀌어도 동작과 무관 하다. 하지만 LVS는 매우 민감하다. 회로 전체를 살펴보고 '오류(error)', '경고(warning)', '통보(notice)' 등을 알려 줄 수 있으면 좋겠지만 Netgen은 그정도 지능(intelligence)은 가지고 있지 않다. 최근 상용 반도체 설계 도구에 인공지능(AI) 넣기가 한창이다. 오픈-소스 도구에도 이의 적용이 멀지 않았다.



이어서 두 네트리스트 사이에 하위회로 구성이 다르다는 점을 지적하고 있다. 회로도에 없는 SWITCH2X1이 레이아웃에 존재한다. Netgen은 하위회로 레이아웃을 상위회로로 평활화(flatten) 하여 등가성을 확인해 준다. 두 회로의 전체 구성(사용된 트랜지스터의 수, 네트의 수)은 일치하지만 네트 연결에 불일치가 발견 되었다는 보고다.

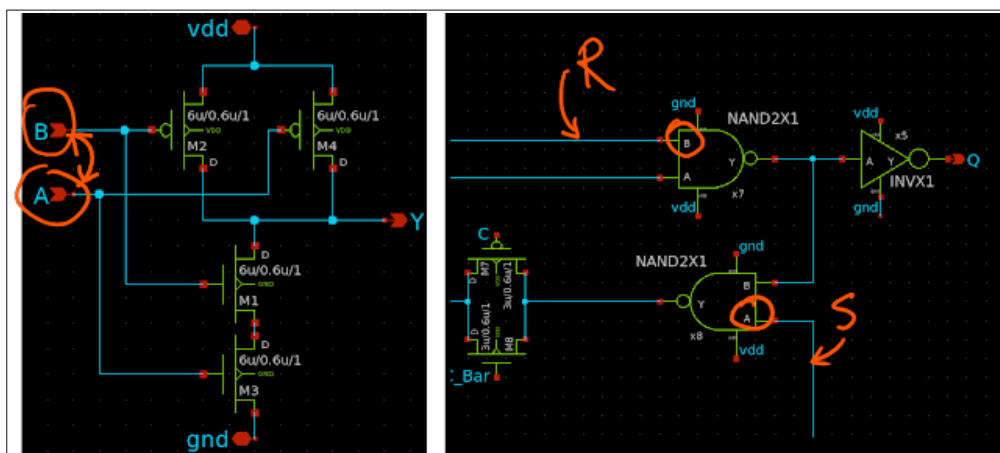
```

43 Device classes INVX1 and INVX1 are equivalent.
44 Flattening unmatched subcell SWITCH2X1 in circuit DFFSR74 (0)(2 instances)
45
46 Subcircuit summary:
47 Circuit 1: DFFSR74 | Circuit 2: DFFSR
48 -----|-----
49 pfet (4) | pfet (4)
50 nfet (4) | nfet (4)
51 NAND2X1 (4) | NAND2X1 (4)
52 INVX1 (4) | INVX1 (4)
53 Number of devices: 16 | Number of devices: 16
54 Number of nets: 16 | Number of nets: 16
55 -----|-----
56 NET mismatches: Class fragments follow (with fanout counts):
57 Circuit 1: DFFSR74 | Circuit 2: DFFSR
58 -----|-----
59
60 Net: R | Net: R
61 NAND2X1/B = 2 | NAND2X1/B = 1
62 | | NAND2X1/A = 1
63 | |
64 Net: S | Net: S
65 NAND2X1/B = 1 | NAND2X1/B = 2
66 NAND2X1/A = 1 |
67

```

레이아웃 DFFSR74 회로에서 네트 R이 두개의 NAND게이트 모두 B 포트에 연결되었으나 DFFSR의 회로도에서는 각각 B와 A 포트에 연결되어 있다. 네트 S의 경우 DFFSR74에서는 각각 A와 B에 연결되었지만 DFFSR 회로도는 모두 B 포트에 연결 되었다. NAND 게이트는 A와 B 포트가 바뀌어도 기능상 등가이지만 LVS의 불일치를 바로 잡도록 하자. 레이아웃을 수정하기 보다 회로도를 수정하는 편이 수월하다. 회로도의 수정은 다음과 같다.

- 하위회로 NAND2X1의 A와 B를 맞바꾼다.
- 네트 R이 X7의 B에, 네트 S는 X8의 A에 연결 한다.



회로도 수정 후 LVS를 실시하여 두 네트리스트가 일치하는 결과를 얻었다.

```

Contents of circuit 1: Circuit: 'DFFSR74'
Circuit DFFSR74 contains 16 device instances.
  Class: pfet             instances:  4
  Class: NAND2X1          instances:  4
  Class: nfet             instances:  4
  Class: INVX1            instances:  4
Circuit contains 16 nets.
Contents of circuit 2: Circuit: 'DFFSR'
Circuit DFFSR contains 16 device instances.
  Class: pfet             instances:  4
  Class: NAND2X1          instances:  4
  Class: nfet             instances:  4
  Class: INVX1            instances:  4
Circuit contains 16 nets.

Circuit 1 contains 16 devices, Circuit 2 contains 16 devices.
Circuit 1 contains 16 nets,   Circuit 2 contains 16 nets.

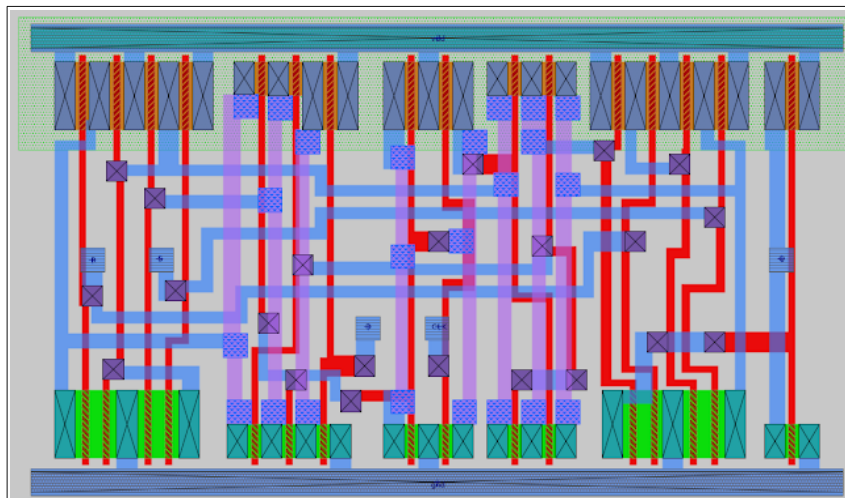
Final result:
Circuits match uniquely.
    
```

← LVS success!

회로도를 수정하였으므로 시뮬레이션을 다시 실시하여 반드시 검증해야 함은 물론이다.

V. 표준 셀 D-플립 플롭 레이아웃 평가

설계에 사용될 부품을 규격화 해 놓음으로써 얻는 장점이 있다. 배치와 배선을 수행하는 자동화 도구에 의해 활용되려면 부품의 외형 규격화(standardized footprint)는 필수다. 배선 자동화 도구가 사용할 금속층의 규칙으로 금속 2층은 수직배선, 금속 3층은 수평배선으로 규정했다고 하자. 전역 배선 규정은 LEF(Library Exchange Format)에 명시된다. 이 규정으로는 위에서 작성한 DFFSR74는 매우 부적절하다. 셀 내부에 유휴공간이 많고 수평배선으로 금속2가 다수 사용되었기 때문에 전역배선의 공간이 없다. 제대로 그린 표준 셀 DFFSR의 레이아웃은 아래와 같다. 약 32개 가량의 트랜지스터를 포함하는 레이아웃은 하위 셀을 불러오지 않고도 충분히 최적화된 셀을 그릴 수 있다.



표준 셀은 레이아웃 뿐만 아니라 자동 배치와 배선 도구를 위한 정보를 제공해야 한다. 표준 셀의 크기와 배치 시 상하 좌우 뒤집힘(flip)의 가능 여부, 핀(포트)의 금속층 좌표와 입출력 방향 그리고 금속 층별 배선을 금지하는 영역(obstruction)을 정의해 주어야 한다. 자동 배선도구는 셀 내부를 들여다 보지 않는다. 따라서 OBS 영역은 셀 내부에서 사용된 금속층의 영역을 표시하여 배선에 사용하지 못하도록 한다. 표준 셀 DFFSR의 LEF 매크로 정의는 아래와 같다.

```
MACRO DFFSR
    CLASS CORE ;
    FOREIGN DFFSR ;
    ORIGIN 0.000 0.000 ;
    SIZE 69.000 BY 39.000 ;
    SYMMETRY X Y ;
    SITE core ;

    PIN D
        DIRECTION INPUT ;
        USE SIGNAL ;
        PORT
            LAYER metal2 ;
            RECT 27.900 14.850 29.100 17.100 ;
        END
    END D
    .....
    PIN Q
        DIRECTION OUTPUT ;
        USE SIGNAL ;
        PORT
            LAYER metal2 ;
            RECT 63.900 15.900 65.100 18.150 ;
        END
    END Q
    PIN vdd
        DIRECTION INOUT ;
        USE POWER ;
        SHAPE ABUTMENT ;
        PORT
            LAYER metal1 ;
            RECT -0.900 37.800 69.900 40.200 ;
        END
    END vdd
    PIN gnd
        .....
    END gnd
    OBS
        LAYER metal1 ;
        RECT 1.200 30.900 3.000 37.500 ;
        .....
        RECT 66.000 1.500 67.800 5.100 ;
```

```

LAYER via1 ;
  RECT 30.900 27.300 33.000 29.400 ;
  .....
LAYER metal2 ;
  RECT 15.900 31.800 18.900 33.900 ;
  RECT 19.800 31.800 21.900 33.900 ;
  .....
END
END DFFSR

```

오픈-소스 Magic 도구는 레이아웃으로부터 표준 셀의 라이브러리 정보를 추출해준다. “내 칩 MPW 디자인 킷 [2]”은 레이아웃에서 LEF 정보의 추출과 통합 수정될 경우 두 레이아웃 사이의 네트리스트 비교(LVL, Layout-vs-Layout)등 일련의 표준 셀 구축용 스크립트를 제공한다. 자신만의 최적화된 표준 셀 라이브러리를 구축해 보자.

-
- [1] KION My Chip MPW, <http://mpw.kion.or.kr/>
 - [2] “내 칩 MPW” 디자인 킷, <https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK>
 - [3] SkyWater130 Process, <https://skywater-pdk.readthedocs.io/en/main/rules/assumptions.html>
 - [4] 플립-플롭, <https://ko.wikipedia.org/wiki/%ED%94%8C%EB%A6%BD%ED%94%8C%EB%A1%AD>
 - [5] Flip-Flop, [https://en.wikipedia.org/wiki/Flip-flop_\(electronics\)](https://en.wikipedia.org/wiki/Flip-flop_(electronics))
 - [6] CDx4HCT74 Dual D-Type Positive-Edge-Triggered Flip-Flops With Clear and Preset, <https://www.ti.com/lit/gpn/CD74HCT74>
 - [7] 장효과 트랜지스터,
https://ko.wikipedia.org/wiki/%EC%9E%A5%ED%9A%A8%EA%B3%BC_%ED%8A%B8%EB%9E%9C%EC%A7%80%EC%8A%A4%ED%84%B0
 - [8] Field Effect Transistor, https://en.wikipedia.org/wiki/Field-effect_transistor
 - [9] Predicting CMOS Speed with Gate Oxide and Voltage Scaling and Interconnect Loading Effects, https://www.researchgate.net/publication/3063052_Predicting_CMOS_speed_with_gate_oxide_and_voltage_scaling_and_interconnect_loading_effects
 - [10] LEF/DEF Reference Manual, <https://www.ispd.cc/contests/18/lefdefref.pdf>