

ASIC 설계를 위한 CAD 기술

徐仁煥, 金春慶
LG 半導體(株) DA 센타

I. ASIC 설계와 CAD 기술

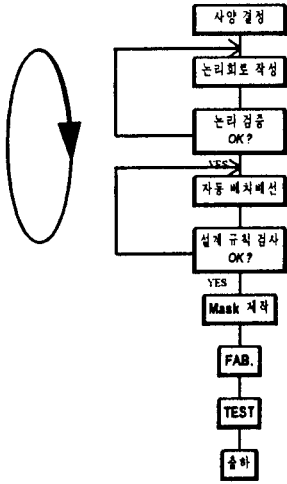
최근 가격 파괴의 경쟁이 모든 산업 전반에서 형성되고 있다. 특히 전기 전자 제품 및 computer 관련 제품의 가격인하는 세인의 이목을 집중시키고 있다. 이와같은 가격 경쟁의 원동력은 제품의 원가 경쟁력 확보이다. 따라서 많은 수의 부품을 줄이고 보다 싼 가격에 다양한 기능을 가지는 ASIC(Application Specific IC), ASSP(Application Specific Standard Product) 제품 개발 경쟁이 치열하다. 또한 이들 제품의 수요기간이 갈수록 단축되어 빠른 제품 개발만이 무한 경쟁에서 생존할 수 있다. 이와같이 고집적 ASIC이 단기간에 개발 가능한 이유는 ASIC CAD 기술의 발전에 있다.

본문에서는 ASIC을 설계하기 위하여 사용되고 있는 software(이하 tool)의 기능을 소개하고 이를 구현하기 위한 알고리즘을 고찰하였다. 따라서 일반적으로 사용되고 있는 논리회로 작성을 통한 ASIC 개발 과정과 CAD tool을 고찰하고 새로운 경향인 논리 합성을 통한 ASIC 개발과정을 고찰하였다. 또한 현재 활발하게 논의되고 있는 deep sub micron 기술에 따른 ASIC 설계 tool의 영향에 대하여 기술하였다. 본문에서 사용되는 용어는 참고 문헌에서 자세한 내용을 얻을 수 있다.

II. ASIC 설계과정과 CAD tool

ASIC 설계는 미리 준비된 표준 논리 소자(이하 cell)를 사용하여 설계한다. 각 cell은 설계 과정에 따라 다양한 형태로 표현된다. 그러나 그 본질은 동일하며 단지 표현의 방법만 다를 뿐이다. 따라서 논리회로 설계를 하는 과정에서는 symbol 형태로 표현되고 실제 IC로 구현될 때는 기하학적인 도형의 집합으로 표현된다.

최근까지 사용된 ASIC 설계 과정은 그림 1에서와 같다. 먼저 개발하고자 하는 제품의 사양을 결



(그림 1) ASIC 설계과정

정하고 사양에 의하여 논리회로를 작성한 후, 모의 실험(이하 simulation)을 통하여 논리가 검증된다. Simulation이 완료된 단계까지를 전단부 설계(이하 front-end 설계)라 한다. Front-end 설계는 사양을 사람이 쉽게 접근할 수 있도록 graphic으로 논리를 표현하고 검증하는 과정이다. Front-end 설계가 완료되면 이를 실제 IC로 제조하기 위한 후단부 설계(이하 back-end 설계)를 한다. Back-end 설계의 첫 단계는 Layout을 설계하는 것이다. Layout은 마치 사진에서 피사체를 렌즈에 옮기는 과정과 같다. 논리회로에서 각 cell에 해당하는 layout cell을 배치하고 연결도의 의해서 배치된 cell을 배선한다. 완성된 layout의 오류를 검사하고 mask를 제작한다. Mask는 사진에서 필름에 해당된다. 따라서 mask 제작과정은 필름을 현상하는 과정에 해당된다. 필름을 인화지에 놓고 인화하듯이 mask를 실리콘위에 두고 인화를 한다. 이 과정이 제조 과정이다. 제조가 된 IC는 불량품을 선별하기 위해 test 과정을 거친 후 양품만 출하된다. 이와같은 ASIC 설계 과정에서 필요로 하는 CAD tool의 특징과 이를 구현하기 위한 알고리즘을 설계 단계별로 기술하였다.

1. 논리회로 편집기(이하 schematic capture)
논리 cell에 해당하는 symbol을 이용하여 논리

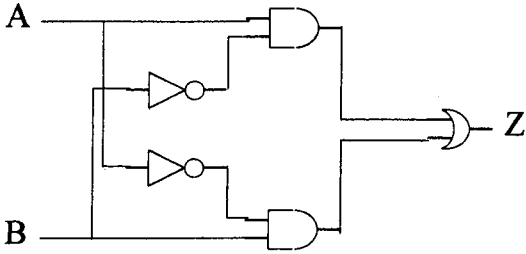
회로도를 그려주는 일종의 graphic editor이다. 목적은 설계 사양을 사람이 이해하기 쉽게 그림으로 표현하고 이를 simulation 할 수 있도록 연결도를 출력하는 것이다. 그러나 일반적인 graphic editor가 bitmap 정보를 한정된 평면에서만 편집가능하도록 한 것과는 달리 무한 평면에서 편집하도록 하며 vectored data를 관리한다. 논리 회로도는 연결도에 따라 논리 symbol과 symbol을 wire로 연결한다. 이때 연결을 net이라 하고 net의 집합을 netlist라 하며 각각의 symbol을 instance라 한다. 그림 2에서 논리 symbol을 보였고 그림 3에서는 논리 symbol을 이용하여 XOR 논리 회로도를 보였었다. 그림 3에서 I1, I2, ... I5는 각 instance name이고 N1, N2, ... N5는 net name이다.

Schematic capture의 성능은 net에 따른 instance의 상호 연결 상태를(이하 connectivity) 잘 유지하는 data structure를 구성 하는데 있다. 따라서 instance를 추가하거나 삭제할 때 data structure에서 즉시 connectivity를 수정할 수 있어야 한다. 이미 graph 이론에서 node를 추가하거나 삭제하는 알고리즘이 널리 알려져 있다. 이에 따라 schematic capture tool은 hash table과 linked list의 적절한 조합으로 data structure를 구성한다. 물론 graphic editor이므로 clipping, rotate 등의 graphic operation 기능도 있어야 한다.

한편 다양한 schematic capture가 제각기 다른 방법으로 netlist를 표현하므로 상호 정보 교류에 큰 지장을 주게 되었다. 이를 해결하기 위하여 IEEE에서 EDIF(Electronic Data Interchange Format)이라는 표준 format을 제정하여 현재 대부분의 schematic tool이 이를 입출력할 수 있다.

논리 기능	Symbol
Not	
2 input And	
2 input Or	

(그림 2) 논리 symbol

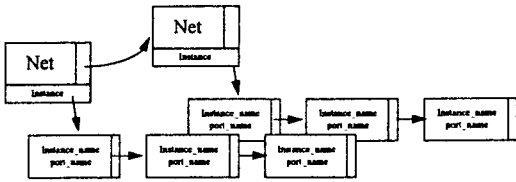


〈그림 3〉 Schematic diagram

```

struct net {
    char *net_name;
    CONNECT *next;
    NET *next;
} NET;

struct connect{
    char *inst_name
    char *port_name;
    CONNECT *next;
} CONNECT
    
```



〈그림 4〉 Data structure for schematic captur

```

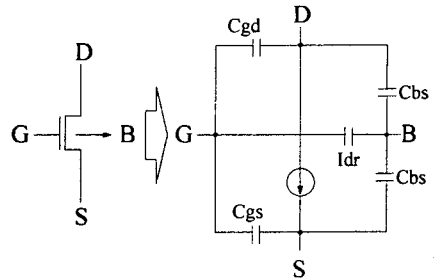
(edif LGS
(status
(edifversion 2 0 0)
(ediflevel 0)
(written
(written
(timestamp 1995 9 15 19 9)
)
)
(external pad-library pla-library)
(library new_version
(cell top-cell
(viewmap ....)
(view masklayout real-geometry
(interface .....
(contents .....
)
)
)
)
)
)
    
```

〈그림 5〉 EDIF 예제

그림 5에서는 EDIF의 예제를 보였다. 한편 논리 회로도도 없고 netlist만 있는 경우 이를 역으로 논리 회로도 변환하는 schematic generator가 개발되었다. 그러나 현재까지의 결과는 사람이 이해하기 힘든 결과가 생성되어 효용가치가 떨어진다.

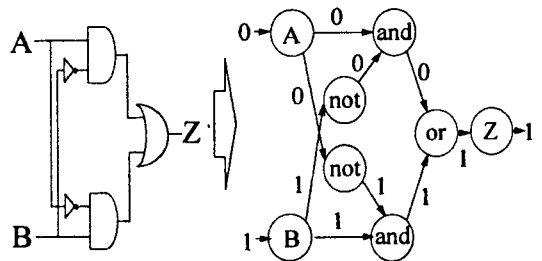
2. 논리 검증(이하 simulation)

Simulation에는 여러가지가 있다. Circuit, logic, timing, fault 등 다양한 목적을 위하여 simulator가 개발되었다. Circuit simulation은 트랜지스터와 저항 및 capacitor 등의 소자로 구성된 회로를 시간 또는 주파수에 따른 전압과 전류의 변화도를 simulation하는 것이다. 이때 대량의 행렬 계산을 고속으로 연산하는 방법이 사용된다. 주로 analog 회로를 검증할 때 사용된다. 그림 6에서는 circuit simulation을 위한 트랜지스터의 modeling을 보였다.



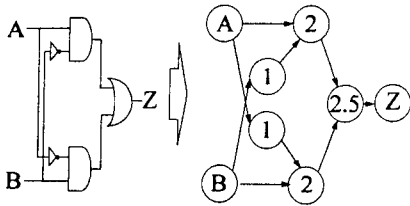
〈그림 6〉 트랜지스터의 modeling

Logic simulation은 그림 7에서와 같이 netlist를 connectivity graph로 표현한다. 각 node에서는 논리 연산을 하고 그 결과를 edge에 둔다. Logic simulation은 circuit simulation과는 달리 모든 시간에 대하여 모든 node를 계산하지 않아도 된다. 도미노 게임처럼 값이 변경된 시간에서 변경된 부분부터 시작되어 이 값이 공급되는 부분에 대하여서만 논리 값을 계산하므로 연산 시간이 단축된다. 이와같은 기법을 event-driven이라 한다.

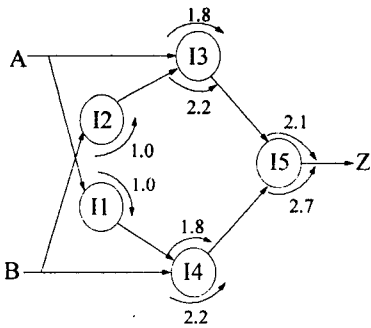


〈그림 7〉 Connectivity graph

Timing simulation은 특정 node에서 node까지 도달하는데 걸리는 시간을 계산하거나 전체에서 가장 많은 시간 이 소요되는 경로를 추출하는 등 시간에 대한 simulation을 하는 것이다. 그림 8에서와 같이 주로 graph의 trace 알고리즘을 사용하여서 최장/최단 소요 시간 경로를 추출하고 이때 소요된 시간을 구한다. 이렇게 구하여진 시간은 setup/hold time violation 등 시간에 관련된 논리 오류를 검사하는데 사용된다. 그림 8에서 A에서 Z까지 도달하는데 최고 5.5, 최소 4.5의 시간이 소요된다. 최근에는 각 cell에서 조차 pin-to-pin 경로에 따른 시간이 달라 보다 정확한 논리 오류를 검사하기 위하여 그림 9에서와 같이 복잡한 graph가 사용되기도 한다. 그림 9에서는 A-I1-I4-I5-Z의 경로에서 5.5의 최장시간이 소요된다.



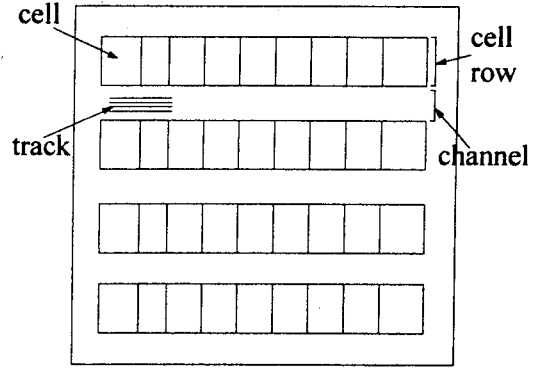
〈그림 8〉 Timing delay graph



〈그림 9〉 Pin-to-Pin delay가 고려된 timing graph

3. 자동 배치 배선

ASIC의 back-end 설계는 그림 10에서와 같이 동일한 높이를 가지는 cell들을 서로 인접하게 위치시켜서 power 선을 공유하게 한다. 이러한 이웃한 cell의 열을 cell row라 한다. Cell row와

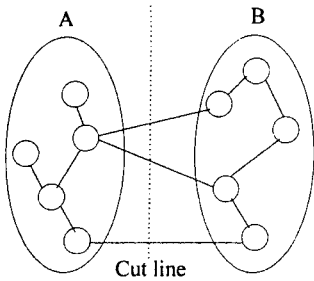


〈그림 10〉 자동 배치 배선

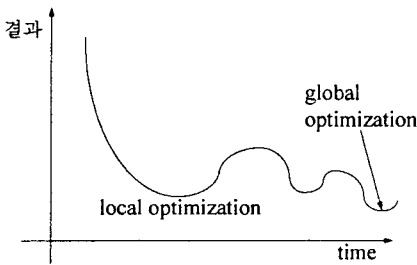
row사이에 공간을 확보하여 배선을 하도록 한다. 이 배선 영역을 channel이라 한다. Channel은 실제 배선이 이루어지는 track의 집합으로 구성되어 있다. 따라서 cell을 임의의 cell row중에서 임의의 위치에 배치시키는 과정을 placement라 하고 placement가 완료된 cell을 연결도에 따라 연결하는 과정을 routing이라 한다.

1) Placement

Cell 배치에 있어서 최선의 방법은 존재하지 않는다. 단지 설계자의 경험에 의한 최적의 배치방법이 많이 연구되어 왔다. 가장 많이 사용되는 알고리즘으로 min-cut과 simulated annealing 방법이 있다. Min-cut placement는 그림 11에서 같다. Cell을 node로 표현하고 net을 edge로 표현한다. 그리고 전체를 집합 A, B 2개로 임의로 분할한다. 그리고 A와 B사이에 cut-line을 두고 cut-line을 통과하는 edge의 수가 감소하는 방향으로 A, B의 cell을 상호 교환한다. 더 이상의 감소가 이루어지지 않게되면 집합 A를 다시 AA와 AB로 분리하여 같은 방법을 되풀이 한다. 이러한 과정을 한 집합에 임의의 cell만 존재할 때까지 계속한다. 분할이 완료되면 각 집합에 있는 cell을 실제 위치에 mapping한다. Min-cut 알고리즘의 특징은 cut-line을 통과하는 edge의 숫자를 최소화 하여 이웃하는 집합사이의 배선이 최소가 되어 전체 크기가 최소가 되도록 하는 것이다. Min-cut 알고리즘은 비교적 빠른 시간내에 배치를 완성시켜 준다.



〈그림 11〉 Min-cut 알고리즘



〈그림 12〉 Simulated annealing에서 time vs. result.

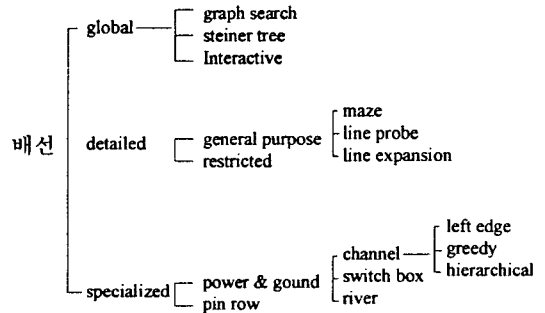
Simulated annealing 알고리즘의 착안은 금속을 고온 열처리하는 원리에 있다. 금속을 고온 상태에서 서서히 온도를 감소시키면 금속 원자가 무질서한 불안정 상태에서 안정된 상태로 되는 과정을 다음과 같이 modeling 하였다. Cell을 임의로 위치에 배치하여 무질서한 불안정 상태(배선이 복잡한 상태)에서 상호 위치 교환을 통하여 전체 배선복잡도가 감소되는 안정된 상태로 되는 것이다. 그림 12에서 수행 시간에 따른 결과를 보였다. 최초에 발생하는 안정된 상태에서(local optimization) 시간이 지나면 약간 불안정한 상태와 안정된 상태가 반복되어 가장 안정된 상태인 global optimization에 도달한다. 따라서 min-cut과 같은 종류의 알고리즘이 local optimization 결과를 가지는데 비하여 더 좋은 결과를 얻을 수 있다. 그러나 수행 시간이 있어서는 보다 많은 시간을 요한다. 왜냐하면 온도를 감소시키는 정도를 너무 빠르게 하면 좋은 결과를 얻을 수 없기 때문이다. 따라서 최근의 경향은 최초에 min-cut과 같은 종류의 알고리즘으로 local optimization에 빠르게 도달한 후 simulated annealing을 사용하여 global optimization에 도달

하게 하여 전체 수행 시간을 줄인다.

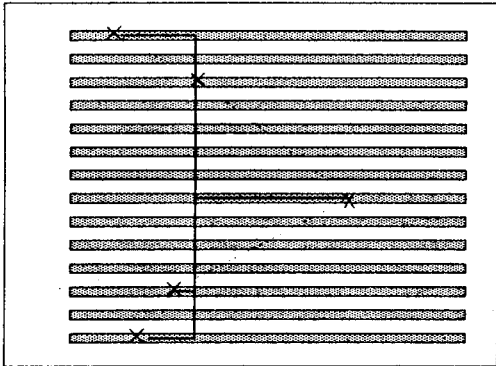
2) Routing

배선 문제는 그림 13에서와 같이 상황에 따라 적절한 알고리즘이 개발되어 있다. 배선은 배치가 완료된 상태에서 layout에서 cell row에 따른 channel의 모양을 정하고 routing 할 channel의 순서를 결정한다. 배선의 결과는 channel의 모양과 순서에 많은 영향을 받는다. Channel 결정이 끝나면 각 net별로 전체영역에서 개략적인 연결모양을 결정한다. 이를 global routing이라 한다. 그림 14에서는 steiner tree를 이용한 global routing 방법을 보였다. 연결되어야 할 좌표의 x축 중간값을 설정하고 이를 기준으로 $y=Xmid$ 직선을 긋는다. 그리고 연결해야할 각 점에서 $y=Xmid$ 로 사상을 한다. 그림 14에서 실선으로 표현된 것이 global routine의 결과이다. Global routing의 목적은 layout에서 배선의 복잡도를 분산 시키고 각 channel에서 필요한 routing 정보를 결정하는 것이다.

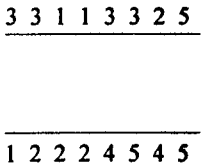
Channel routing 알고리즘 중에서 LEA(Left Edge Alogrithm)은 그림 15에서와 같다. Channel 연결정보 중에서 가장 왼쪽부터 시작되는 net부터 track을 차지하는 길이를 구한다(그림 16). 예제의 경우 5개의 net에 대하여 필요한 track의 길이를 구하였다. 만약 왼쪽에서 시작되는 것이 여러 개 있을 경우 (그림 16의(1)에서 1, 3번net) track의 길이가 긴/짧은 것부터 차례로 정렬한다. 그리고 그림 16-(2)에서와 같이 서로 중복되지 않도록 track을 할당한다. 그림 17에서 그림 15



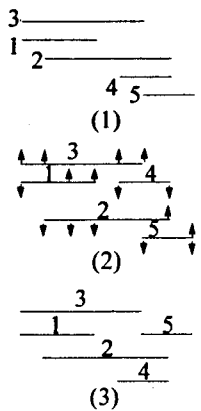
〈그림 13〉 Routing algorithms



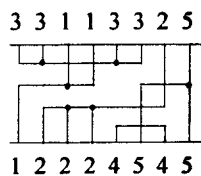
〈그림 14〉 Global routing



〈그림 15〉 Routing 정보



〈그림 16〉 Track 할당



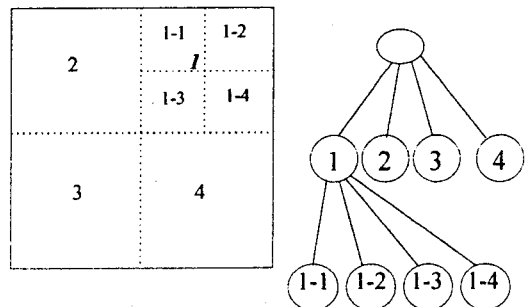
〈그림 17〉 Channel routing 결과

에 대한 channel routing한 결과를 보였다. 이와같이 channel routing의 주요 과제는 필요한 track의 숫자를 최소화 하여 각 channel의 높이를 최소화하여 전체 면적을 줄이는 것이다.

4. Layout Editor

Layout Editor는 가장 초기에 개발된 CAD tool이다. 목적은 layout을 편집하는 기능을 제공하는 것이다. Schematic capture와 같이 일종의 도형 편집기이며, 무한 평면에서 vectored data를 편집한다. 특히 layout editor에서 처리되는 도형의 갯수는 천문학적 숫자이므로 이를 편집 가능하도록 하는 능력과 특정 도형을 빠르게 찾는 능력이 주요한 요소이다. 그림 18에서는 layout editor의 data structure인 quad-tree를 보였다. Layout 평면을 4개의 영역으로 분리하고 각 사분면을 다시 4개의 영역으로 분리한다. 이러한 분리를 한 영역에 임계 갯수의 도형만 있을 때까지 계속한다. Quad-tree의 주요 기술로는 전체 tree의 depth를 조절하여 speed를 향상시키는 것과 각 사분면에 동시에 걸쳐있는 도형을 처리 방법이 있다.

Layout된 결과에 대한 기록 형식이 모든 tool마다 상이하나 이 부분에 있어서는 이미 GDS II (Calma format)이 업계 표준이 되었고 CIF (Caltech Intermediate Format)이 학계 표준이 되어 있다. 따라서 대부분의 tool이 두 가지 모두에 대한 입출력 기능을 제공하고 있다.

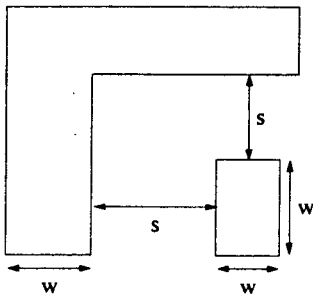


〈그림 18〉 Quad-tree

5. DRC(Design Rule Check)

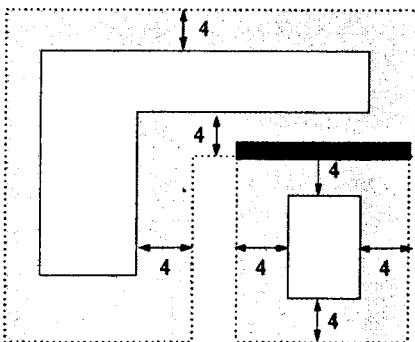
Layout은 공정 기술에 의하여 설정된 기하학적

인 설계 규칙(Design Rule)을 따라야 한다. 설계 규칙은 각 도형이 가져야 할 최소의 크기와 이웃하는 도형 사이의 최소 간격에 대하여 규정하고 있다. 그림 19에서 간단한 예제를 보였다. 따라서 DRC는 설계규칙에 적합하게 layout이 작성되었는지 여부를 검사하는 tool이다. 그 방법으로 도형 연산이 사용된다. 만약 예제에서 최소 거리가 8이라고 하면 이에대한 설계규칙 검사방법은 그림 20에서와 같다. 각 도형을 사방으로 4(최소 거리의 절반)만큼 확장한다(Oversize). 그리고 두 도형에 대하여 AND를 한다. 그러면 그림 20에서와 같이 짙게 표시된 사각형만 남는다. 따라서 이 부분이 최소 거리 규칙을 위반한 영역이 된다.



s : space rule
w: width rule

〈그림 19〉 설계 규칙

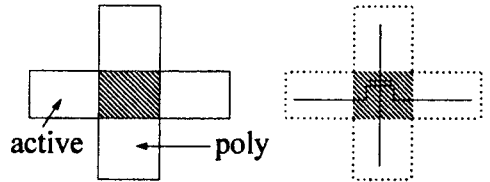


〈그림 20〉 설계 규칙 검사 알고리즘

6. LVS(Layout vs. Schematic)

LVS는 최초 확인 완료된 논리 회로도가 layout

으로 변경되는 과정에서 오류가 있었는지 여부를 검사하는 것이다. 먼저 layout에서 connectivity를 추출하여 연결도 graph를 생성하고 schematic의 netlist에서 연결도 graph를 생성한 후, 두 graph에서 최외각에서부터 pattern match를 하여 상호 연결도 비교를 한다. 그림 21에서는 layout에서 회로도를 추출하는 일반적 방법에 대하여 보였다. MOS는 active와 polysilicon이 교차 되는 부분에서 생성된다. 나머지는 connection wire로 처리된다. MOS를 추출하는 방법은 active에 해당하는 도형과 polysilicon에 해당하는 도형에 대하여 XOR한 결과 도형이 MOS가 된다. 추출된 도형 이외의 부분은 wire로 처리된다(wire = (polysilicon OR active) MINUS MOS).



〈그림 21〉 MOS 추출과정

III. ASIC 설계 방법과 CAD tool의 변화

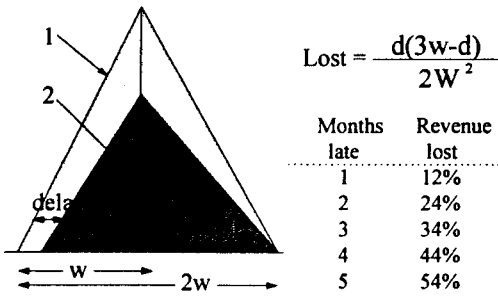
최근 들어서 ASIC 설계와 관련된 주변 여건의 변화가 극심하였다. ASIC의 생명인 설계기간 단축에 대한 요구가 커진 반면에 집적도는 향상되어 설계자로 하여금 四面楚歌에 시달리게 했다. 한편 multimedia을 중심으로 ASIC 시장이 확대되어 analog logic 사용이 커짐에 따라 논리 검증에 어려움이 발생되었다. 또한 deep sub micron 기술에 진입하여 기존에 무시되던 short channel의 물리적 현상에 대하여 주의를 기울이게 되었다.

1. Time to Market

ASIC의 가장 큰 장점은 빠른 설계에 있다. 그러나 최근에는 제품의 시장 수명이 단축되어, 제품 개발 지연이 매출에 막대한 손실을 초래한다. 그림

22에서는 이에 대한 현상을 보이고 있다. 그림에서 시장 진입이 5개월 지연될 경우 54%의 손실이 발생됨을 알 수 있다. 따라서 논리 회로도 작성을 통한 설계 기간의 단축이 한계에 도달하자 새로운 방법이 창출되었다. 설계자는 설계 사양을 특정한 논리 기술 언어로(HDL : Hardware Description Language) 작성하면 simulation될 뿐 아니라 netlist와 논리 회로도들을 자동으로 생성된다. 이를 logic synthesis라 한다. 이와 같은 방법이 top-down 설계다. Logic synthesis는 HDL을 2level boolean equation으로 변환시킨 후 이를 최적화한다(Optimization). 그런 후 실제 사용되어질 cell로 mapping 되어 최종 netlist가 생성된다.

현재 사용되고 있는 HDL로는 VHDL(Very high speed integrated circuit Hardware Description Language)과 Verilog-HDL이 있다. 그림



1. Revenue Curve for on time
2. Revenue Curve for delayed Market
W : Market window

〈그림 22〉 Time to Market

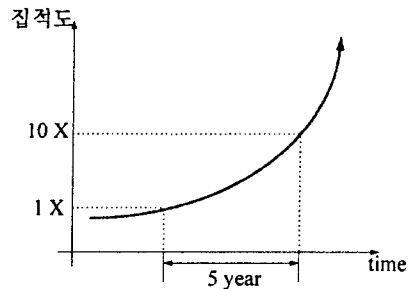
```
process(RESET,ENABLE,GATED_CLK,Z)
begin
  if(RESET = '1') then
    Z <= 0;
  elsif (GATED_CLOCK`event and
    GATED_CLK = '1') then
    if(Z = 7) then
      Z <= 0;
    else
      Z <= Z+1;
    endif
  endif
end process
```

〈그림 23〉 VHDL 예제

23에서 VHDL의 예제를 보였다.

2. Increase Design Complexity

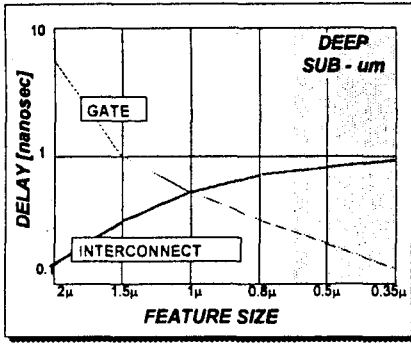
그림 24에서와 같이 ASIC의 집적도는 매 5년마다 10배가 증가되고 있다. 따라서 DRC, LVS에서 처리시간도 기하학적으로 증가되어 사용에서 현실성을 잃게 될 것이다. 또한 이에 따른 하드웨어의 요구량도 커서 전체 비용이 증가된다. 이와같은 문제를 해결하기 위하여 계층적 검사방법이 활발하게 개발되어 상용 tool이 시판되고 있다. 계층적 검사는 ASIC에서 동일한 cell이 반복적으로 사용되므로 이를 한번만 검사하고 나머지는 검사하지 않아도 되는 점에 착안한 것이다. 따라서 서로 다른 계층에 연결되어 있는 도형에 대한 처리 기법이 중요한 요소 기술이다.



〈그림 24〉 집적도의 증가

3. Introduce deep sub micron technology

Sub micron 기술은 집적도의 향상에 따른 문제점 이외에 문제시되지 않았던 물리적 현상에 주목하게 한다. 그 중 한가지가 배선에 따른 delay이다. 이전의 기술에서는 wire delay는 주요한 요소가 아니었다. 그러나 현재는 wire delay가 그림 25에서와 같이 전체 delay에서 주요 요소로 등장하였다. 따라서 설계 초기에 wire delay를 예측하는 방법이 중요하게 되었다(Pre-simulation). 또한 layout에서 wire delay를 추출하여 simulation으로 feedback 되도록 하는 과정도 중요하게 되었다(Back annotation, Post-simulation).



〈그림 25〉 Wire delay

IV. 결 론

ASIC 설계에서 필요한 CAD tool을 구현하기 위한 알고리즘을 고찰하였다. 대부분의 CAD tool 알고리즘은 graph 이론과 matrix 연산으로 구성된다. 특히 graph에서 특정 node의 탐색과 편집 및 path trace 기술이 주요 요소 기술임을 인지하였다.

한편 deep sub micron의 가속화는 고집적에 따른 system on silicon을 가능하게 하지만 power, digital 소자의 analog 특성화, wire delay, layout에서 cross-talk 현상 등의 문제를 야기시키고 있다. 따라서 CAD tool은 이 현상들을 최소화하거나 정확하게 추출하는 방향으로 개선/개발되고 있다. 그러나 개별 tool의 성능 못지않게 단기간에 고집적의 ASIC을 개발할 수 있는 설계 방법론 (Design Methodology)에 대한 활발한 연구가 있어야 할 것이다

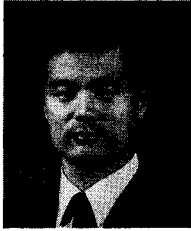
참 고 문 헌

[1] Carver Mead, Lynn Conway, "Introduc-

tion to VLSI Systems", Addison-Wesley Publishing Co.

- [2] Nel Weste, Kamran Eshraghian, "Principles of CMOS CVLSI Design A Systems Perspective", Addison-Wesley Publishing Co.
- [3] Steven M. Rubin, "Computer Aids for VLSI Design", Addison-Wesley Publishing Co.
- [4] Lance A. Glasser, Daniel W. Dobberpuhl, "The Design and Analysis of VLSI Circuits", Addison-Wesley Publishing Co.
- [5] Bryan Preas, Michael Lorenzetti, "Physical Design Automation of VLSI Systems", The Benjamin/Cummings Publishing Co.
- [6] Amar Mukherjee, "Introduction to nMOS & CMOS VLSI Systems Design", Prentice-Hall.
- [7] Daniel D. Gajski, Nikil D. Dutt, Allen CH Wu, Steve Y-L Lin, "HIGH-LEVEL SYNTHESIS INTRODUCTION TO CHIP AND SYSTEM DESIGN", Kluwer Academic Publishers.
- [8] Carl Sechen, "VLSI Placement and Global Routing using Simulated Annealing", Kluwer Academic Publishers.
- [9] Guy Rabbat "HANDBOOK OF ADVANCED SEMICONDUCTOR TECHNOLOGY AND COMPUTER SYSTEMS", VAN NOSTRAND REINHOLD.
- [10] Donald E. Carter, Barbara Stiwell Baker, "CE Concurrent Engineering The product Development Environment for the 1990s", Addison-Wesley Publishing Co.
- [11] 이귀로, 박송배, 전주식, 경종민, 임인철, "VLSI 설계를 위한 CAD 알고리즘", 방한출판사

저 자 소 개



金 春 慶

1952年 1月 4日生
 1974年 2月 서울대학교 전기공학과 학사 졸업
 1981年 7月 IOWA대학교 전자 및 컴퓨터 공학과 석사 졸업
 1984年 9月 Minnesota대학교 전자공학과 박사 졸업

1974年 5月~1979年 7月 국방과학원 선임연구원
 1979年 8月~1981年 9月 IOWA大 RA 및 TA
 1981年 10月~1984年 9月 Minnesota大 RA 및 TA
 1983年 6月~1983年 9月 Metheus社 VLSI 연구 스텝
 1984年~1989年 GS社 중앙연구소 VLSI 연구 스텝
 1989年~현재 LG 반도체(주) 기술연구소 연구위원

주관심 분야 : Design Methodology 개발



徐 仁 煥

1965年 2月 13日生
 1987年 2月 경북대학교 전자공학과 학사 졸업
 1989年 2月 경북대학교 전자공학과 석사 졸업

1989년 1月~현재 LG 반도체(주) 선임연구원

주관심 분야 : Placement & Routing, High level synthesis, Design re-use